

Grado en Matemáticas

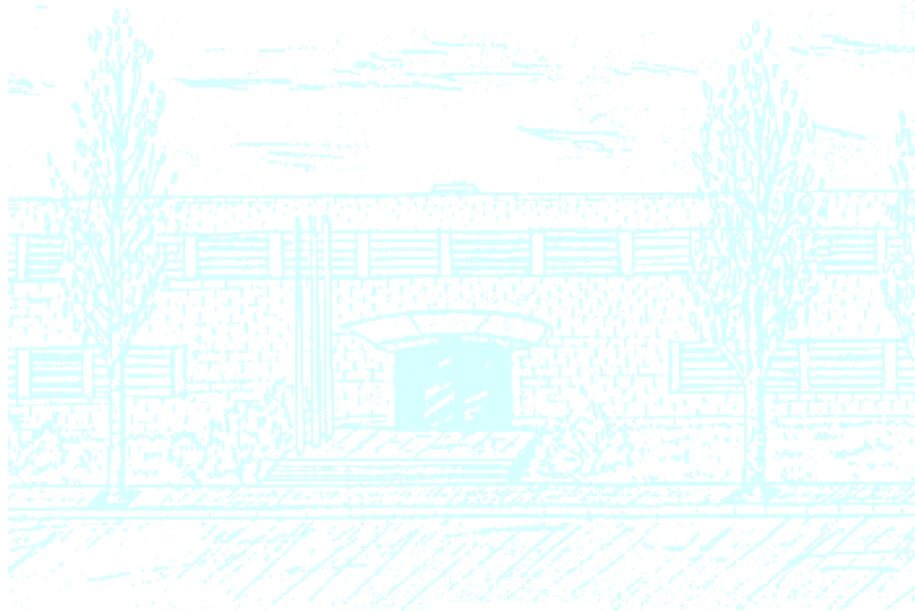
Título: Transformada de Fourier: aplicaciones al procesamiento de señales

Autor: Carlos María Rodríguez

Director: Santiago Boza Rocho

Departamento: Departamento de Matemáticas

Convocatoria: 2016/2017



Resumen

Palabras clave: Series de Fourier, Transformada de Fourier, Teoría de la Señal.

La transformada de Fourier, en sus numerosas variantes, es una herramienta clave para el análisis en frecuencia de funciones o señales. Junto a la serie de Fourier, supuso una revolución matemática, tanto desde el punto de vista teórico como del práctico. Este trabajo pretende realizar una introducción teórica a los resultados del análisis de Fourier, y mostrar algunas de sus posibles aplicaciones.

Las primeras secciones serán dedicadas a presentar las series de Fourier, la descomposición en funciones trigonométricas de señales periódicas. Tras ello, veremos la transformada discreta de Fourier, una técnica análoga a usar cuando sólo tenemos un conjunto de muestras en lugar de la función completa. Cerraremos este bloque con la transformada rápida de Fourier, una implementación de la discreta mucho menos costosa.

Pasaremos después a ver la transformada de Fourier usual: un operador que permite calcular el espectro de funciones no necesariamente periódicas. Se trata de una extensión continua de la serie de Fourier a señales generales, con ciertas limitaciones pero muy potente. Lo último que veremos en la sección teórica será la base matemática del muestreo de señales, y el diseño de filtros.

Cerraremos el documento con una serie de aplicaciones de todo lo visto, ejemplos que permiten analizar y modificar señales de sonido e imágenes a partir de su contenido en frecuencia.

Contenido

1. Introducción	5
-----------------------	---

Contenido Teórico

2. Funciones Trigonométricas.....	6
3. Series de Fourier	8
4. Convergencia Puntual	11
5. DFT, Transformada Discreta de Fourier.....	16
6. FFT, Transformada Rápida de Fourier	20
7. La Transformada de Fourier	22
8. Teorema de Shannon.....	28
9. Filtros Digitales.....	35

Contenido Práctico

10. Sonido	41
11. Edición Imagen	50

Bibliografía.....	55
--------------------------	-----------

Anexo

A. La Integral de Lebesgue	57
B. Código.....	60

1. Introducción

A lo largo de la historia de las matemáticas, distintas figuras han trabajado con la idea de aproximar o representar funciones como suma de funciones trigonométricas. Desde mediados del siglo XVIII se tiene constancia de distintos ejemplos de interpolación trigonométrica, y ya aparecían métodos para desarrollar en senos o cosenos. De hecho, en 1805 Gauss utilizó un algoritmo muy similar a la mucho más moderna *transformada rápida de Fourier* (¡que no sería desarrollado hasta los 60!)¹, pero pasó desapercibido por estar en un manuscrito sin publicar.

Con el auge de problemas asociados a ecuaciones diferenciales, y el desarrollo de nuevas técnicas, se avanzó notablemente en este campo. Bernouilli y Euler introdujeron la idea de representaciones trigonométricas de funciones; y figuras como d'Alembert, Lagrange o Gauss presentaron series trigonométricas como soluciones a distintas ecuaciones.

Sería en 1807 cuando Joseph Fourier publicaría “*Mémoire sur la propagation de la chaleur dans les corps solides*”. Aquí se introducía por primera vez el concepto de la *Serie de Fourier*, con las que representar cualquier función periódica como serie trigonométrica. A pesar de cierta crítica por parte del comité supervisor por la falta de rigurosidad o generalidad en ciertos aspectos, las ideas introducidas fueron clave. Con la serie llegaría también la *Transformada de Fourier*, una transformación continua al plano de frecuencias y válida para funciones no periódicas.

A lo largo de las siguientes décadas, sus resultados serían refinados y ampliados hasta llegar al concepto moderno del análisis de Fourier y la teoría que lo rodea. Dirichlet formularía en 1828 condiciones necesarias para que una función tenga transformada de Fourier; y con aportaciones de multitud de matemáticos, el conjunto de funciones tratables sería ampliado, y los resultados formalizados.

Paralelamente, y especialmente en el último medio siglo, el análisis de Fourier fue cobrando una importancia cada vez mayor en la práctica. Su concepción ya se debía en gran parte a problemas físicos, y con el paso de los años se diseñarían todo tipo de técnicas para aplicarlo.

El objetivo de este documento es establecer una base teórica del análisis de Fourier, suficiente para mostrar algunas de sus aplicaciones. Comenzaremos con un estudio de las funciones trigonométricas y sus propiedades, hasta llegar a la Serie de Fourier. Aquí pasaremos a un aspecto más práctico, con la transformada de Fourier discreta y la transformada de Fourier rápida.

La sección más larga tratará sobre la transformada de Fourier continua, basada en la teoría de la integral de Lebesgue incluida en el anexo. Por último, veremos un estudio matemático de la toma de muestras; y una descripción de los filtros digitales, encargados de operar sobre señales y estrechamente relacionados con la transformada.

El segundo bloque contendrá una serie de posibles aplicaciones prácticas de toda la teoría desarrollada, para sonido e imagen. Aunque sea imposible realizar un estudio exhaustivo de todas las técnicas basadas en esta teoría, podremos entender las ideas principales en las que se basan.

¹ Más al respecto en [10]. Para más información sobre el contexto matemático de Fourier, véase [20].

Contenido Teórico

En este primer bloque introduciremos la teoría sobre la Serie de Fourier, y posteriormente la Transformada, en la que se basan las aplicaciones prácticas. En ambos casos necesitamos introducir conceptos e ideas hasta llegar a los resultados principales.

Aquí no presentaremos todos los resultados, pues son demasiados. Nos ceñiremos a aquellos estrictamente necesarios para presentar los teoremas centrales y las aplicaciones prácticas. En [6] podemos encontrar un estudio en más profundidad, con una estructura similar a la de este documento.

2. Funciones Trigonómicas

Empecemos estudiando los polinomios trigonométricos, la base de la Serie de Fourier.

Definición 2.1- Un polinomio trigonométrico es una función de la forma

$$p(t) = \sum_{n \in I} c_n e^{2\pi i n t / a};$$

donde I es un conjunto finito de enteros, los c_n son coeficientes complejos y a es el período de la función (y de hecho, de cada componente).

Podemos expresarlo como

$$p(t) = \sum_{n=-N}^N c_n e^{2\pi i n t / a};$$

y diremos entonces que es un polinomio trigonométrico de grado menor o igual a N . De forma menos compacta, podemos representarlo también como suma de senos y cosenos:

$$p(t) = c_0 + \sum_{n=1}^N \left(c_n e^{\frac{2\pi i n t}{a}} + c_{-n} e^{-\frac{2\pi i n t}{a}} \right),$$
$$p(t) = \frac{1}{2} a_0 + \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi n t}{a}\right) + b_n \sin\left(\frac{2\pi n t}{a}\right) \right);$$

donde

$$\begin{cases} a_n = c_n + c_{-n} \\ b_n = i(c_n - c_{-n}) \end{cases}, \text{ para } n \geq 0.$$

Las componentes elementales de estos polinomios son de la forma

$$e_n(t) = e^{2\pi i n t / a};$$

y cumplen:

Proposición 2.2- Para los $e_n(t)$ se tiene

$$\int_0^a e_n(t) \bar{e}_m(t) dt = \begin{cases} a & \text{si } n = m \\ 0 & \text{si } n \neq m \end{cases}.$$

Demostración. Es directo operando normalmente; consideremos $n - m = k$:

$$\begin{aligned} \int_0^a e_n(t) \cdot \bar{e}_m(t) dt &= \int_0^a e^{2\pi i n t / a} \cdot e^{-2\pi i m t / a} dt = \int_0^a e^{2\pi i k t / a} dt = \\ &= \begin{cases} \int_0^a e^{2\pi i k t / a} dt = \frac{a}{2\pi i k} \left[e^{\frac{2\pi i k t}{a}} \right]_0^a = 0, & \text{si } k \neq 0 \\ a, & \text{si } k = 0 \end{cases} \quad \blacksquare \end{aligned}$$

Dadas estas definiciones, es natural considerar el siguiente espacio vectorial:

Def. 2.3- T_N es el conjunto de polinomios trigonométricos de grado menor o igual a N .

Dado que cada polinomio viene definido unívocamente por sus coeficientes c_n , se trata de un espacio vectorial de dimensión $2N + 1$.

Prop. 2.4- Podemos equipar a este espacio con el producto escalar definido por

$$(p, q) = \int_0^a p(t) \bar{q}(t) dt,$$

y su norma asociada

$$\|p\|_2 = \sqrt{\int_0^a p(t) \bar{p}(t) dt}.$$

Es decir, las funciones e_n son ortogonales en este espacio y por tanto independientes: componen la base.

Obs. 2.5- Nótese que

$$(p, e_n) = c_n \|e_n\|_2^2 = c_n a.$$

Y por tanto

$$c_n = \frac{1}{a} \int_0^a p(t) e^{-2\pi i n t/a} dt,$$

que son los llamados coeficientes de Fourier.

Aplicando las expresiones para la forma en senos y cosenos:

$$a_n = \frac{2}{a} \int_0^a p(t) \cos\left(\frac{2\pi n t}{a}\right) dt, n \geq 0,$$

$$b_n = \frac{2}{a} \int_0^a p(t) \sin\left(\frac{2\pi n t}{a}\right) dt, n \geq 0.$$

Con estas fórmulas podemos calcular los coeficientes directamente para cualquier función trigonométrica. Esto es, podemos realizar su análisis espectral sin necesidad de más resultados.

Dado que p es periódica, las integrales para los coeficientes pueden calcularse en cualquier intervalo de tamaño a . Así, si tomamos $(-a/2, a/2)$, tenemos

$$p \text{ par} \Leftrightarrow c_{-n} = c_n, n \in \mathbb{Z} \Leftrightarrow b_n = 0, n \in \mathbb{N}$$

$$p \text{ impar} \Leftrightarrow c_{-n} = -c_n, n \in \mathbb{Z} \Leftrightarrow a_n = 0, n \in \mathbb{N}$$

Esto se deduce considerando que el seno es antisimétrico y el coseno simétrico, utilizando las expresiones anteriores.

Obs. 2.6- Calculando la norma de p tenemos

$$\|p\|_2^2 = a \sum_{n=-N}^N \sum_{m=-N}^N c_n \bar{c}_m (e_n, e_m).$$

Y con la ortogonalidad de la base, nos queda la identidad de **Parseval**:

$$\frac{1}{a} \int_0^a |p(t)|^2 dt = \sum_{n=-N}^N |c_n|^2.$$

3. Series de Fourier

Dada una función cualquiera $f: \mathbb{R} \rightarrow \mathbb{C}$ de periodo a , la cuestión central es si podemos encontrar una descomposición de f de la forma

$$f(t) = \sum c_n e^{2\pi i n t / a}.$$

En 1807, *Fourier* demostró que sí, pero sólo si consideramos sumas infinitas. Por suerte, disponemos de herramientas matemáticas más potentes que Fourier que facilitarán el desarrollo, aunque debemos ser rigurosos. En las próximas páginas introduciremos los conceptos necesarios para demostrar los distintos resultados.

Comencemos suponiendo que f cumple

$$\int_0^a f(t)^2 dt < +\infty;$$

esto es, que la integral existe y es finita. Estudiaremos de momento este caso y más tarde intentaremos generalizar.

Def. 3.1- Consideremos el conjunto de funciones periódicas

$$L_p^2(0, a) = \left\{ f: \mathbb{R} \rightarrow \mathbb{C} \mid \int_0^a |f(t)|^2 dt < +\infty, f \text{ de periodo } a \right\}.$$

Este conjunto es un espacio vectorial complejo con el producto por escalar y la suma de funciones, y obsérvese que T_N es un subespacio suyo. Que la suma de dos funciones se mantiene en el conjunto no es evidente, y debe ser demostrado:

$$\begin{aligned} f, g \in L_p^2(0, a); \quad f + g \text{ tiene periodo } a, y \\ \int_0^a |f(t) + g(t)|^2 dt &= \int_0^a (f + g)(\overline{f + g}) dt = \int_0^a (|f|^2 + |g|^2 + 2\operatorname{Re}(f \cdot \bar{g})) dt \\ &\leq \int_0^a 2(|f|^2 + |g|^2) dt = 2 \int_0^a |f|^2 dt + 2 \int_0^a |g|^2 dt < +\infty \quad \blacksquare \end{aligned}$$

La función f con la que vamos a trabajar pertenece por tanto a $L_p^2(0, a)$. Podemos definir aquí el producto escalar

$$(f, g) = \int_0^a f(t) \bar{g}(t) dt$$

y su norma asociada

$$\|f\|_2 = \sqrt{(f, f)} = \left(\int_0^a |f(t)|^2 dt \right)^{1/2}.$$

Es necesario observar que esta norma de $f \in L_p^2(0, a)$ puede ser 0 aunque f no sea siempre nula; por ejemplo si es nula en un conjunto finito de puntos. Así, esta pareja de producto y norma no lo son formalmente con la igualdad usual. Es necesario introducir el siguiente concepto:

Def. 3.2- Dos funciones $f, g \in L_p^2(0, a)$ son iguales *casi por todo* (c.p.t.) si

$$\int_0^a |f(t) - g(t)| dt = 0.$$

Así, considerando la igualdad casi por todo sí tenemos que el producto escalar anterior es un verdadero producto, e igualmente la norma.

Volviendo a la cuestión central, dijimos que en principio no podemos expresar $f \in L_p^2(0, a)$ como suma finita de funciones trigonométricas. Es natural preguntarse cuál sería el polinomio trigonométrico en T_N que mejor la aproxime, esto es, los coeficientes x_n que minimicen

$$\left\| f - \sum_{n=-N}^N x_n e_n \right\|_2.$$

Sea $p(t) = \sum_{n=-N}^N x_n e_n$. Desarrollemos la expresión (su cuadrado, por simplicidad):

$$\|f - p\|_2^2 = \|f\|_2^2 + \|p\|_2^2 - 2 \cdot \operatorname{Re}(f, p).$$

Sabemos por Parseval que

$$\|p\|_2^2 = a \sum_{n=-N}^N |x_n|^2, \quad y \quad (f, p) = \sum_{n=-N}^N \bar{x}_n (f, e_n).$$

Si introducimos $c_n = 1/a \cdot (f, e_n)$ y sustituimos, tenemos

$$\begin{aligned} \|f - p\|_2^2 &= \|f\|_2^2 + a \sum_{n=-N}^N (|x_n|^2 - 2 \operatorname{Re}(\bar{x}_n c_n)) = \\ &= \|f\|_2^2 + a \sum_{n=-N}^N (|c_n - x_n|^2 - |c_n|^2). \end{aligned}$$

Y por tanto el mínimo se alcanza en $c_n = x_n$. En resumen:

Teo. 3.3- Existe un único polinomio trigonométrico f_N en T_N que minimiza $\|f - f_N\|_2$. Viene dado por

$$f_N(t) = \sum_{n=-N}^N c_n e_n(t),$$

donde

$$c_n = \frac{1}{a} \int_0^a f(t) e^{-2i\pi n t/a} dt.$$

Además, si consideramos de nuevo $\|f - p\|_2^2 = \|f\|_2^2 + a \sum_{n=-N}^N (|c_n - x_n|^2 - |c_n|^2)$;

Prop. 3.4- Introduciendo $c_n = x_n$ y f_N en la expresión previa obtenemos:

$$a \sum_{n=-N}^N |c_n|^2 + \|f - f_N\|_2^2 = \|f\|_2^2.$$

Y, como consecuencia,

$$a \sum_{n=-N}^N |c_n|^2 \leq \frac{1}{a} \int_0^a |f(t)|^2 dt;$$

conocida como la Desigualdad de Bessel.

Aquí es donde entra la condición de que f pertenezca a L_p^2 , pues tenemos

$$\sum_{n=-N}^N |c_n|^2 < \infty,$$

y en consecuencia, $c_n(f) \rightarrow 0$ si $|n| \rightarrow \infty$.

Lo natural es preguntarse ahora por el comportamiento de f_N cuando $N \rightarrow \infty$.

Teo. 3.5- Dada $f \in L_p^2(0, a)$, se tiene que f_N tiende a f en $L_p^2(0, a)$ cuando $N \rightarrow \infty$. Dicho de otro modo,

$$\int_0^a |f(t) - f_N(t)|^2 dt \rightarrow 0, \text{ si } N \rightarrow \infty.$$

Este teorema nos lleva a

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i n t / a},$$

donde los c_n son los *coeficientes de Fourier* de f . Obsérvese que esta igualdad es en el espacio $L_p^2(0, a)$; no nos dice que ambas expresiones valgan lo mismo para cada t . Formalmente, este resultado implica que la familia $(e_n)_{n \in \mathbb{Z}}$ es base de $L_p^2(0, a)$, y que la serie del lado derecho es sumable a f en este espacio.

Además, ahora podemos también expandir la identidad de Parseval a un resultado más general. En concreto, si $f \in L_p^2(0, a)$

$$\sum_{n=-\infty}^{\infty} |c_n|^2 = \frac{1}{a} \int_0^a |f(t)|^2 dt.$$

Podemos demostrar también con este teorema que si dos funciones tienen los mismos coeficientes de Fourier, entonces son iguales en casi todas partes. Esto es:

Prop. 3.6- Sean $f, g \in L_p^2(0, a)$;
 $c_n(f) = c_n(g) \forall n \in \mathbb{Z} \Rightarrow f = g \text{ c.p.t.}$

Demostración. Por linealidad, nos basta con ver si para $f \in L_p^2(0, a)$,

$$c_n(f) = 0 \forall n \in \mathbb{Z} \Rightarrow f = 0 \text{ c.p.t.}$$

Aplicando la identidad de Parseval,

$$\frac{1}{a} \int_0^a |f(t)|^2 dt = \sum_{n=-\infty}^{\infty} |c_n|^2 = 0,$$

y

$$\frac{1}{a} \int_0^a |f(t)|^2 dt = 0 \Rightarrow f = 0 \text{ c.p.t. en } (0, a) \blacksquare$$

Con esto concluimos que existe una relación bidireccional entre coeficientes de Fourier y funciones, pero sólo en el sentido de igualdad de $L_p^2(0, a)$.

4. Convergencia Puntual

Hemos demostrado el resultado en L_p^2 , pero querríamos verlo para funciones más generales y encontrar una mejor convergencia; nos interesaría poder usar la serie de Fourier como igual a la función en cada punto. Consideremos el conjunto de funciones siguiente:

Def. 4.1- Definimos el espacio vectorial

$$L_p^1(0, a) = \left\{ f: \mathbb{R} \rightarrow \mathbb{C} \mid \int_0^a |f(t)| dt < +\infty, f \text{ tiene periodo } a \right\}.$$

Nótese que este conjunto contiene a $L_p^2(0, a)$, intentaremos generalizar el Teorema 3.5 aquí.

Obs. 4.2- Dado que los coeficientes vienen dados por

$$c_n = \frac{1}{a} \int_0^a f(t) e^{-2i\pi n t/a} dt,$$

éstos existirán si y sólo si f es integrable en $(0, a)$, pues una propiedad de la integral de Lebesgue² es que f es integrable en un intervalo I si y sólo si su módulo es integrable.

Así, podemos considerar para una f en $L_p^1(0, a)$ su serie de Fourier

$$\sum_{n=-\infty}^{\infty} c_n(f) e^{2\pi i n t/a}.$$

Un primer resultado importante es que, igual que en $L_p^2(0, a)$, tenemos $c_n(f) \rightarrow 0$ cuando $n \rightarrow \infty$.

Teo. 4.3- Teorema de Riemann-Lebesgue: Dado un intervalo (a, b) acotado, f integrable en (a, b) , entonces la integral

$$I_n = \int_a^b f(x) e^{2i\pi n x} dx \rightarrow 0, \text{ si } |n| \rightarrow \infty.$$

Demostración. Supongamos primero que f tiene derivada continua en $[a, b]$. Integrando por partes, tenemos

$$I_n = \frac{1}{2i\pi|n|} [f(x) e^{2i\pi n x}]_a^b - \frac{1}{2i\pi|n|} \int_a^b f'(x) e^{2i\pi n x} dx;$$

y podemos acotar su módulo

$$|I_n| = \frac{1}{2\pi|n|} \left(|f(a)| + |f(b)| + \int_a^b |f'(x)| dx \right).$$

Así, $|I_n| \xrightarrow{|n| \rightarrow \infty} 0$ en este caso. Para demostrarlo para cualquier función, usamos que el conjunto de funciones con derivada continua en $[a, b]$ es denso en $L_p^1[a, b]$, un resultado que no veremos aquí.

² En el anexo A hablaremos sobre la integral de Lebesgue.

Dado un $\varepsilon > 0$, $\exists g_\varepsilon \in C^1([a, b])$ tal que

$$\int_a^b |f(x) - g_\varepsilon(x)| dx \leq \frac{\varepsilon}{2},$$

y por tanto

$$|I_n| \leq \int_a^b |f(x) - g_\varepsilon(x)| dx + \left| \int_a^b g_\varepsilon(x) e^{2i\pi n x} dx \right|.$$

Por el primer resultado,

$$\exists N > 0 \text{ tq } \left| \int_a^b g_\varepsilon(x) e^{2i\pi n x} dx \right| < \frac{\varepsilon}{2};$$

y así, $|n| \geq N \Rightarrow |I_n| \leq \varepsilon$, lo cual demuestra el teorema. ■

Recordemos

$$f_N(t) = \sum_{n=-N}^N c_n e_n(t).$$

Nuestro objetivo es encontrar ahora algún resultado sobre la convergencia puntual de esta serie. Lamentablemente, vamos a necesitar acotar más el conjunto de funciones con el que trabajamos, pues de hecho hay varios resultados que podríamos esperar pero que son falsos en $L_p^1(0, a)$:

- i. $f_N \rightarrow f$ en la norma $L_p^1(0, a)$.
- ii. $f_N(t) \rightarrow f(t)$ casi por todo t .
- iii. Si f es continua en \mathbb{R} , $f_N(t) \rightarrow f(t) \forall t \in \mathbb{R}$.

De hecho, existen funciones en $L_p^1(0, a)$ tales que el límite de $f_N(t)$ diverge para todo t .

Supondremos pues que $f: [a, b] \rightarrow \mathbb{C}$ es continua a trozos. Esto nos asegura que en $[a, b]$ f está acotada y es integrable. Será conveniente para una demostración posterior tener una expresión para la diferencia entre f y f_N ; suponemos $f \in L_p^1(0, a)$ y un t_0 tal que $f(t_0 +)$ y $f(t_0 -)$ existen. Partimos de la definición reordenada

$$f_N(t_0) = \frac{1}{a} \int_{-a/2}^{a/2} \left(\sum_{n=-N}^N e^{2\pi i n \frac{t_0 - x}{a}} \right) f(x) dx,$$

obtenida al combinar las expresiones para f_N y c_n y aprovechando que la función tiene periodo a (podemos integrar sobre cualquier intervalo de tamaño a). Si ahora usamos que

$$\sum_{n=-N}^N e^{2\pi i n t} = \frac{\sin(\pi t(2N + 1))}{\sin(\pi t)},$$

tenemos

$$f_N(t_0) = \frac{1}{a} \int_{-a/2}^{a/2} \frac{\sin\left(\pi \frac{t_0 - x}{a} (2N + 1)\right)}{\sin\left(\pi \frac{t_0 - x}{a}\right)} f(x) dx,$$

y con un cambio de variable

$$f_N(t_0) = \frac{1}{a} \int_{-a/2}^{a/2} \frac{\sin\left(\pi \frac{x}{a} (2N + 1)\right)}{\sin\left(\pi \frac{x}{a}\right)} f(x + t_0) dx.$$

Dado que el integrando tiene periodo a , podemos seguir integrando en $[-a/2, a/2]$. Si cambiamos x por $-x$ en $[-a/2, 0]$, tenemos

$$f_N(t_0) = \frac{1}{a} \int_0^{a/2} \frac{\sin\left(\pi \frac{x}{a} (2N+1)\right)}{\sin\left(\pi \frac{x}{a}\right)} [f(x+t_0) + f(x-t_0)] dx.$$

Aquí usamos el caso concreto $f = 1$, donde $f_N = 1$ si la calculamos. Por tanto,

$$\frac{1}{a} \int_0^{a/2} \frac{\sin\left(\pi \frac{x}{a} (2N+1)\right)}{\sin\left(\pi \frac{x}{a}\right)} dx = \frac{1}{2}.$$

Sea $y_0 = \frac{1}{2}[f(t_0+) + f(t_0-)]$, obtenemos

$$f_N(t_0) - y_0 = \frac{1}{a} \int_0^{a/2} \frac{\sin\left(\pi \frac{x}{a} (2N+1)\right)}{\sin\left(\pi \frac{x}{a}\right)} [f(x+t_0) + f(x-t_0) - 2y_0] dx \blacksquare$$

Con esta expresión, veamos uno de los teoremas centrales de ese tema.

Teo. 4.4- Teorema de Dirichlet: Sea $f \in L_p^1(0, a)$. Si los límites $f(t_0+), f(t_0-)$ existen para un cierto t_0 , y ambas derivadas también existen, entonces

$$f_N(t_0) \rightarrow \frac{1}{2}[f(t_0+) + f(t_0-)]; \text{ si } N \rightarrow \infty.$$

Demostración. De que las derivadas existan sabemos que

$$\frac{1}{x}(f(t_0+x) - f(t_0+)) \text{ y } \frac{1}{x}(f(t_0-x) - f(t_0-))$$

tienden a límites finitos cuando $x \rightarrow 0$. Así, lo mismo ocurre para

$$\varphi(x) = \frac{f(t_0+x) - f(t_0+) + f(t_0-x) - f(t_0-)}{\sin(\pi x/a)}.$$

Por tanto, existen $\alpha, M > 0$ tales que $|\varphi(x)| \leq M \forall x \in (0, \alpha]$. Dado que $f \in L_p^1(0, a)$, φ es integrable en $[\alpha, a/2]$ y

$$|\varphi(x)| \leq M + |\varphi(x)|\chi_{[\alpha, a/2]}(x) \quad \forall x \in (0, a/2]$$

Una propiedad de la integral de Lebesgue es que

$$\left. \begin{array}{l} |g(x)| \leq h(x) \quad \forall x \in I \\ h \text{ integrable por Lebesgue en } I \end{array} \right\} \Rightarrow g \text{ integrable por Lebesgue en } I.$$

Así, φ es integrable en $(0, a/2)$ y por el Teorema de Riemann-Lebesgue (4.3) tenemos que

$$f_N(t_0) - y_0 = \frac{1}{a} \int_0^{a/2} \varphi(x) \sin\left((2N+1)\frac{x}{a}\right) dx \xrightarrow{N \rightarrow \infty} 0 \blacksquare$$

Con esto tenemos la convergencia de una serie de Fourier localmente, y que sólo depende del entorno del punto. Podemos ver ahora un resultado global.

Def. 4.5- Decimos que una función f es de variación acotada en $[a, b]$, $f \in BV[a, b]$, si

$$\exists M \text{ tq. } \sum_{k=0}^n |f(t_{k+1}) - f(t_k)| \leq M$$

para cualquier subdivisión $a = t_0 < t_1 < \dots < t_{n+1} = b$, donde $n \in \mathbb{N}$ es arbitraria.

Teo. 4.6- Sea $f: \mathbb{R} \rightarrow \mathbb{C}$ de periodo a y de variación acotada. Tenemos

- i) $\forall t_0 \in \mathbb{R}, f_N(t_0) \rightarrow \frac{1}{2}[f(t_0 +) + f(t_0 -)];$ si $N \rightarrow \infty$.
- ii) Si además f es continua en $[\alpha, \beta]$, f_N converge uniformemente a f en $[\alpha, \beta]$.

Veamos un último teorema sobre convergencia uniforme.

Teo. 4.7- Sea f de periodo a , continua en \mathbb{R} , diferenciable en $[0, a]$ excepto en un número finito de puntos y que f' es continua a trozos. Tenemos

- i) La serie de Fourier de f' se obtiene derivando término a término la de f .
- ii) Los coeficientes de Fourier de f cumplen

$$\sum_{n=-\infty}^{\infty} |c_n(f)| < +\infty.$$

- iii) La serie de Fourier de f converge uniformemente a f en \mathbb{R} .

Demostración. Con las hipótesis dadas tenemos que $c_n(f)$ puede ser calculada por partes, y que f' está en $L_p^2(0, a)$ por ser continua a trozos.

- i) Integrando por partes tenemos:

$$c_n(f) = \frac{1}{a} \int_0^a f(t) e^{-2\pi i n t/a} dt = \frac{1}{2\pi i n} \int_0^a f'(t) e^{-2\pi i n t/a} dt.$$

Ya que f tiene periodo a y es continua. Así, los coeficientes de Fourier de f' son

$$c_n(f') = \frac{2\pi i n}{a} c_n(f).$$

- ii) Deducimos el segundo punto de la desigualdad a partir de esta expresión, y usando que $(a + b)^2 \geq 2ab$:

$$|c_n(f)| = \frac{a}{2\pi |n|} |c_n(f')| \leq \frac{a}{4\pi} \left(\frac{1}{n^2} + |c_n(f')|^2 \right).$$

Por tanto, $\sum_{n=-\infty}^{\infty} |c_n(f)| < +\infty$ ya que $f' \in L_p^2(0, a)$.

- iii) Para el último resultado, observemos que $\sum_{n=-\infty}^{+\infty} |c_n(f)| < +\infty$ implica que (f_N) es una sucesión de Cauchy con la norma uniforme en $[0, a]$, y por tanto converge uniformemente a una función g continua. Dado que converge también en $L_p^2(0, a)$, por la unicidad del límite nos queda que $f = g$ casi por todo. Dado que ambas son continuas, $g = f$ ■

Obs. 4.8- Cuanto más regular sea una función, esto es, cuantas más derivadas continuas tenga, más rápido tenderán a 0 sus coeficientes de Fourier. Podemos ver esto integrando repetidamente por partes, como en la demostración anterior.

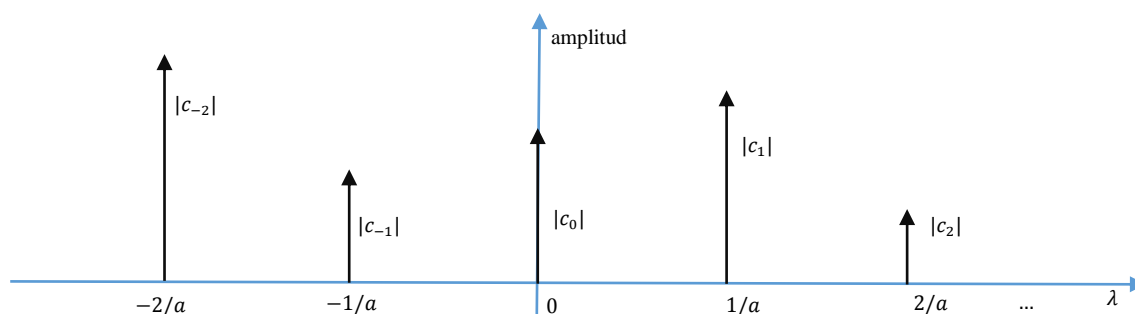
Terminamos la sección con una serie de conceptos y una conclusión.

Def. 4.9- Dada f función de periodo a , con serie de Fourier asociada

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2i\pi n t/a},$$

el *espectro* de f es el conjunto de parejas $\left(\frac{n}{a}, c_n\right), n \in \mathbb{Z}$.

Una forma común de representar el espectro de una función mediante esquemas como este:



El *espectro de amplitudes* contiene líneas regularmente separadas $1/a$, cuyas alturas representan la amplitud asociada a cada frecuencia. Para $|n| = 1$ tenemos las líneas correspondientes a las *frecuencias fundamentales*; el resto son las de los *armónicos*.

Igualmente podemos formar el *espectro de fases*, donde las parejas representadas son $\left(\frac{n}{a}, \theta_n\right), n \in \mathbb{Z}$; donde $c_n = |c_n|e^{i\theta_n}, \theta_n \in [-\pi, \pi)$. Nos referiremos a estos valores como *fases* o *ángulos*.

Es conveniente notar una serie de resultados sobre cómo la función f puede afectar a sus coeficientes, como hicimos con los polinomios trigonométricos:

- a) f real $\Leftrightarrow c_{-n} = \bar{c}_n, n \in \mathbb{Z} \Leftrightarrow a_n, b_n$ reales, $n \in \mathbb{N}$
- b) f par $\Leftrightarrow c_{-n} = c_n, n \in \mathbb{Z} \Leftrightarrow b_n = 0, n \in \mathbb{N}$
- c) f impar $\Leftrightarrow c_{-n} = -c_n, n \in \mathbb{Z} \Leftrightarrow a_n = 0, n \in \mathbb{N}$
- d) f real, par \Leftrightarrow la sucesión (c_n) es real y par

Nótese que las implicaciones hacia el lado izquierdo son en el sentido de *casi por todo*; por ejemplo, $c_{-n} = c_n$ implica que $f(t) = f(-t)$ c.p.t.

La serie de Fourier, en resumen, descompone una señal periódica en las frecuencias que la forman. Es importante que en muchos casos el proceso es bidireccional, y a partir del espectro de una señal podemos reconstruirla. Así, se abren gran cantidad de posibilidades tanto para analizar como para operar con ellas. Como veremos, podemos calcular la serie de una señal, modificarla como queramos y deshacer la transformación para obtener la nueva señal deseada.

5. DFT, Transformada Discreta de Fourier

Lamentablemente, muchas veces no es factible calcular los coeficientes como hemos visto, pues en lugar de tener acceso a todos los valores de la función, tan sólo tenemos una serie de muestras. Esto es especialmente corriente en la práctica, y necesitamos nueva teoría para paliarlo. Por suerte existe un algoritmo que nos permite calcular aproximaciones en estos casos: la Transformada Discreta de Fourier.

Veamos bajo qué condiciones trabajaremos. Conocemos el periodo a de f , así como una muestra de tamaño N :

$$f\left(k\frac{a}{N}\right) = x_k, \quad k = 0, 1, \dots, N-1.$$

En la práctica las muestras tomadas suelen ser números reales, pero el proceso es análogo con imaginarios. Asumimos también que su serie asociada converge puntualmente a f y que en puntos de discontinuidad

$$f(t) = \frac{1}{2}(f_N(t+) + f_N(t-)).$$

Dado que tenemos N puntos, calcularemos N coeficientes, los que en principio serían más relevantes. Dado que los coeficientes tienden a 0, querremos los de n en torno a 0: $-N/2 \leq n < N/2$. Intentaremos calcularlos con dos métodos distintos: aproximando la integral e interpolando los puntos.

Estudiemos primero la integral

$$c_n = \frac{1}{a} \int_0^a f(t) e^{-2i\pi nt/a} dt.$$

No podemos operar por tener tan sólo una serie de muestras, pero podemos aproximar el resultado con la regla del trapecio:

$$\frac{1}{a} \int_0^a f(t) e^{-2i\pi nt/a} dt \approx \frac{1}{a} \left(\frac{a}{N} \sum_{k=0}^{N-1} x_k e^{-\frac{2i\pi nk}{N}} \right).$$

Nos queda un primer conjunto de coeficientes

$$c_n^* = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-2i\pi nk/N}, \quad -\frac{N}{2} \leq n < \frac{N}{2}.$$

Intentémoslo ahora interpolando los N puntos. Para ello trataremos las muestras como un vector de N complejos, y buscaremos realizar cierto cambio de base. Consideremos los siguientes conceptos:

Def. 5.1- Dados dos vectores complejos de tamaño N , definimos el producto

$$\langle x, y \rangle = \sum_{k=0}^{N-1} x_k \overline{y_k}.$$

Y la norma asociada

$$||x|| = \sqrt{\sum_{k=0}^{N-1} |x_k|^2}.$$

Def. 5.2- Los vectores normalizados

$$\phi_n = \frac{1}{\sqrt{N}} \left(1, e^{\frac{2\pi i n}{N}}, e^{\frac{2\pi i 2n}{N}}, \dots, e^{\frac{2\pi i kn}{N}}, \dots, e^{\frac{2\pi i (N-1)n}{N}} \right)$$

son los llamados tonos digitales puros de orden N . El conjunto

$$\mathcal{F}_N = \{\phi_n\}_{n=0}^N$$

es la base de Fourier de n puntos.

En esencia, no es más que el conjunto de valores que toman las funciones elementales en cada punto donde tomamos una muestra, con un factor normalizador para simplificar cálculos.

Aquí debemos hacer un comentario importante sobre qué coeficientes calcular. Vemos primero que

$$\phi_{N+n} = \phi_n.$$

Esto es, se trata de un conjunto de vectores que se repiten periódicamente. Por tanto, nos basta con tener N de estos vectores consecutivos para la base, la única diferencia será el orden de los coeficientes calculados. Usamos $n = 0..N$ para simplificar las expresiones. Trataremos esto más al final de la sección.

Prop. 5.3- El conjunto $\{\phi_n\}_{n=0}^N$ es en efecto una base ortogonal en \mathbb{R}^N .

Demostración. Nos basta con ver que son ortonormales, pues serán independientes y por tanto una base. Consideremos el producto de dos vectores del conjunto (multiplicado por N para simplificar):

$$\begin{aligned} N \langle \phi_{n_1}, \phi_{n_2} \rangle &= \left\langle e^{\frac{2\pi i n_1 k}{N}}, e^{\frac{2\pi i n_2 k}{N}} \right\rangle = \sum_{k=0}^{N-1} e^{\frac{2\pi i n_1 k}{N}} e^{-\frac{2\pi i n_2 k}{N}} = \\ &= \sum_{k=0}^{N-1} e^{\frac{2\pi i (n_1 - n_2) k}{N}} = \frac{1 - e^{2\pi i (n_1 - n_2)}}{1 - e^{2\pi i (n_1 - n_2)/N}}. \end{aligned}$$

Suponiendo $n_1 \neq n_2$ obtenemos 0. De forma análoga, gracias al factor normalizador ($\frac{1}{\sqrt{N}}$), vemos que en efecto se trata de una base ortonormal. ■

Queremos encontrar un cambio de base de \mathbb{R} a \mathcal{F}_N ; la representación en esta nueva base del vector de muestras será el conjunto de coeficientes que buscamos (sólo nos quedará suprimir el factor normalizador y reordenar).

Def. 5.4- El cambio de coordenadas de \mathbb{R}^N a \mathcal{F}_N es la llamada Transformada Discreta de Fourier (DFT de sus siglas en inglés). La matriz de cambio de base, F_N , es la matriz de Fourier (de N puntos). Dado un vector $x \in \mathbb{R}^N$, sus coordenadas en la base de Fourier son sus *coeficientes de Fourier*.

Queríamos tener esta matriz en forma explícita.

Partimos del hecho de que el cambio de base inverso sería

$$x = y_0 \phi_0 + y_1 \phi_1 + \dots + y_{N-1} \phi_{N-1} = (\phi_0 \phi_1 \dots \phi_{N-1}) y = F_N^{-1} y;$$

donde y es el vector de coordenadas en la base de Fourier y x las coordenadas en la base usual. Es decir, la matriz del cambio de base inversa tiene como columnas los vectores ϕ_n .

Recordemos que estos vectores son ortonormales por el producto de vectores definido previamente. Así, aunque F_N^{-1} no sea ortogonal en el sentido usual, tenemos que F_N será la matriz traspuesta de una matriz idéntica pero cuyos elementos han sido conjugados (la matriz adjunta, que expresaremos con el superíndice H : F_N^H). De hecho, dado que esta matriz es simétrica, sólo es necesario conjugar. Es decir, abusando ligeramente de notación:

$$F_N = \overline{F_N^{-1}}.$$

Teo. 5.5- La matriz de Fourier de tamaño $N \times N$ tiene como elementos

$$(F_N)_{nk} = \frac{1}{\sqrt{N}} e^{-\frac{2\pi i nk}{N}}, \quad 0 \leq n, k < N.$$

Nótese que en general no se incluiría el factor $1/\sqrt{N}$, sino $1/N$. Aquí lo tenemos por la definición que hemos dado de la base de Fourier, y nos deja una matriz unitaria.

Def. 5.6- Dado $y \in \mathbb{R}^N$, el vector $x = (F_N)^{-1}y$ se conoce como la Transformada Discreta Inversa de Fourier (IDFT) de y .

La DFT y su inversa (IDFT) se pueden expresar también por componentes; siendo x el vector en la base canónica e y su transformada:

$$x_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y_n e^{2\pi i nk/N},$$

$$y_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2\pi i nk/N}.$$

Podemos suprimir el factor normalizador para obtener las expresiones utilizadas normalmente:

$$x_k = \sum_{n=0}^{N-1} y_n e^{2\pi i nk/N}$$

$$y_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-2\pi i nk/N}$$

Obs. 5.7- El vector x de muestras se puede extender periódicamente para que se repita cada N posiciones, por tener f periodo a . Esto, junto a la fórmula anterior, nos hace ver que el vector de su DFT y también es periódico con periodo N .

Nos queda que los coeficientes de Fourier aproximados son:

$$c_n \approx c_n^N = \begin{cases} y_n, & \text{si } 0 \leq n < \frac{N}{2} \\ y_{N+n}, & \text{si } -\frac{N}{2} \leq n < 0 \end{cases},$$

ya que y se extiende periódicamente. Nótese que estos coeficientes son los mismos que obtuvimos aproximando la integral al principio de la sección.

El extender periódicamente los vectores debe hacerse teniendo en mente que los c_n^N calculados son sólo aproximaciones de c_n para $-\frac{N}{2} \leq n < \frac{N}{2}$. Con estas extensiones, tendremos que $c_n^N = y_n \forall n \in \mathbb{Z}$, y los cálculos pueden hacerse sobre cualquier conjunto de N enteros consecutivos. En la práctica se calculan para $n \in [0, N - 1]$ con las fórmulas anteriores.

Merece la pena mencionar el caso en el que las muestras son reales. Por las propiedades ya vistas anteriormente, en este caso tenemos $c_{-n}^N = \overline{c_n^N}$, y por tanto tan sólo necesitamos calcular la mitad de los coeficientes. A partir de los de $0 \leq n < N/2$ podemos conjugar y obtener el resto.

Hemos obtenido un método biyectivo para calcular las aproximaciones de los coeficientes de Fourier a partir de una serie de muestras. En esencia no es más que una transformación lineal de \mathbb{C}^N en sí misma, pero que nos abre una nueva dimensión para trabajar con señales.

6. FFT, Transformada Rápida de Fourier

Lamentablemente, el cálculo de la DFT visto antes es costoso: se compone de $(N-1)^2$ multiplicaciones complejas y $N(N-1)$ sumas; y esto es suponiendo que ya tenemos los valores necesarios procesados y disponibles. En la práctica resulta conveniente si no necesario el uso de algoritmos más rápidos. Para ello la implementación más conocida es la FFT, *Fast Fourier Transform*.

Teo. 6.1- Sea $y = F_N x$ la DFT de x , N par. Para cualquier entero $n \in [0, N/2 - 1]$,

$$y_n = \frac{1}{\sqrt{2}} \left((F_{N/2} x^{(p)})_n + e^{-\frac{2\pi i n}{N}} (F_{N/2} x^{(i)})_n \right),$$

$$y_{N/2+n} = \frac{1}{\sqrt{2}} \left((F_{N/2} x^{(p)})_n - e^{-\frac{2\pi i n}{N}} (F_{N/2} x^{(i)})_n \right).$$

Donde $x^{(p)}, x^{(i)}$ son las secuencias de tamaño $N/2$ que contienen las muestras de posición par e impar, respectivamente.

Demostración: Sea $n \in [0, \frac{N}{2} - 1]$. Separamos la expresión original en pares e impares

$$\begin{aligned} y_n &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2\pi i k n / N} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N/2-1} x_{2k} e^{-4\pi i k n / N} + \frac{1}{\sqrt{N}} \sum_{k=0}^{N/2-1} x_{2k+1} e^{-2\pi i n (2k+1) / N} \\ &= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{N/2}} \sum_{k=0}^{N/2-1} x_{2k} e^{-4\pi i k n / (N/2)} + \frac{1}{\sqrt{2}} \frac{1}{\sqrt{N/2}} e^{-2\pi i n / N} \sum_{k=0}^{N/2-1} x_{2k+1} e^{-4\pi i k n / (N/2)} = \\ &= \frac{1}{\sqrt{2}} \left((F_{N/2} x^{(p)})_n + e^{-\frac{2\pi i n}{N}} (F_{N/2} x^{(i)})_n \right). \end{aligned}$$

Para el otro resultado:

$$\begin{aligned} y_{N/2+n} &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-2\pi i k (N/2+n) / N} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-\pi i k} e^{-2\pi i k n / N} = \\ &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N/2-1} x_{2k} e^{-4\pi i k n / N} - \frac{1}{\sqrt{N}} \sum_{k=0}^{N/2-1} x_{2k+1} e^{-\frac{2\pi i n (2k+1)}{N}} = \\ &= \frac{1}{\sqrt{2}} \left((F_{N/2} x^{(p)})_n - e^{-\frac{2\pi i n}{N}} (F_{N/2} x^{(i)})_n \right). \end{aligned}$$

Donde en el penúltimo paso separamos entre pares e impares, y debido a la exponencial tenemos el signo negativo en los impares. En el último paso repetimos las ideas del primer resultado. ■

También podemos expresar esta relación en forma de matriz por bloques:

$$F_N = \frac{1}{\sqrt{2}} \begin{pmatrix} F_{N/2} & D_{N/2} F_{N/2} \\ F_{N/2} & -D_{N/2} F_{N/2} \end{pmatrix} P_N$$

Donde las matrices son

- $D_{N/2}$ es la matriz diagonal de tamaño $\left(\frac{N}{2}\right) \times \left(\frac{N}{2}\right)$ con entradas $e^{-2\pi i n / N}, n \in [0, \frac{N}{2} - 1]$.
- P_N es la matriz de permutación que transforma un vector x en otro que contiene primero los elementos de posición par y después los de posición impar.

Así, transformamos una DFT de N muestras a dos de tamaño $N/2$. Considerando el crecimiento de la complejidad del cálculo de la DFT, este método hace el proceso notablemente más rápido.

Nos queda un algoritmo recursivo que al llegar a cierto tamaño (número de muestras) dejará de llamarse a sí mismo y utilizará el algoritmo usual para realizar la DFT (introducido en la sección anterior). La elección de este tamaño básico es una discusión de carácter más computacional y por tanto no la trataremos aquí.

En lugar de un coste del orden de N^2 , la FFT crece $O(N \cdot \log_2 N)$. Esta diferencia se hace muy notable en la práctica, considerando la gran cantidad de muestras que suelen tomarse.

Recordemos que si además las muestras son números reales (que suele ser el caso en aplicaciones reales), se tiene $c_{-n} = \bar{c}_n, n \in \mathbb{Z}$. En este caso nos basta calcular una mitad y conjugarlos para obtener todos los coeficientes, ahorrando aún más tiempo.

Disponemos también de un algoritmo rápido para el algoritmo inverso: la IFFT. En forma matricial:

Teo. 6.2- La matriz inversa de Fourier se puede expresar como

$$(F_N)^H = \frac{1}{\sqrt{2}} \begin{pmatrix} (F_{N/2})^H & D_{N/2}(F_{N/2})^H \\ (F_{N/2})^H & -D_{N/2}(F_{N/2})^H \end{pmatrix} P_N,$$

donde $D_{N/2}$ es la matriz diagonal $\begin{pmatrix} N \\ 2 \end{pmatrix} \times \begin{pmatrix} N \\ 2 \end{pmatrix}$ con entradas $e^{2\pi i n/N}, n \in [0, \frac{N}{2} - 1]$.

Nótese que la diferencia entre esta factorización y la anterior radica en el signo de $e^{2\pi i n/N}$. Así, el coste computacional es virtualmente idéntico y se aplican las mismas ideas.

Hay una idea importante que debemos comentar antes de cerrar este apartado. Estos algoritmos se aplican cuando N es par, y si queremos aprovechar la recursividad es preferible que sea una potencia de 2. Así, en la práctica se divide la señal en bloques de tamaño 2^r , r típicamente entre 5 y 10, y aplicamos los algoritmos anteriores para después unir los resultados con diversas técnicas.

La FFT es un algoritmo usado con mucha frecuencia, y vital en aplicaciones que necesiten calcular rápidamente, como retransmisión de datos en directo. Existen muchas variantes para cada situación de la FFT, en las últimas secciones veremos algunas de ellas.

7. La Transformada de Fourier

Hasta ahora hemos estado trabajando con funciones periódicas. Nos gustaría estudiar si existen resultados análogos para funciones en general. En esta sección utilizaremos propiedades de la integral de Lebesgue; en el apéndice A tenemos las definiciones y demostraciones correspondientes.

Def. 7.1- Dada $f \in L^1(\mathbb{R})$, tenemos

$$\mathcal{F} f(\omega) = \hat{f}(\omega) = \int_{\mathbb{R}} e^{-2\pi i \omega x} f(x) dx \quad (7.1)$$

$$\bar{\mathcal{F}} f(\omega) = \int_{\mathbb{R}} e^{2\pi i \omega x} f(x) dx \quad (7.2)$$

Por definición, $\mathcal{F} f$ es la transformada de Fourier de f , y $\bar{\mathcal{F}} f$ es la transformada conjugada, que más tarde veremos que de hecho se trata de la operación inversa, siempre que esté definida. Obsérvese que estas integrales tienen sentido si y sólo si $f \in L^1(\mathbb{R})$, pues $|e^{\pm 2\pi i x \varepsilon}| = 1$.

Teo. 7.2- Teorema de Riemann-Lebesgue: Sea $f \in L^1(\mathbb{R})$, entonces \hat{f} satisface

- i) $\mathcal{F} f$ es continua y acotada en \mathbb{R} .
- ii) \mathcal{F} es un operador continuo de $L^1(\mathbb{R})$ a $L^\infty(\mathbb{R})$ y
$$\|\hat{f}\|_\infty \leq \|f\|_1.$$
- iii) $\lim_{|\omega| \rightarrow +\infty} |\hat{f}(\omega)| = 0$.

Demostración.

i) La continuidad de \hat{f} proviene directamente de la continuidad de la integral 7.1 respecto al parámetro ω . La función $\omega \mapsto e^{-2\pi i \omega x} f(x)$ es continua en \mathbb{R} y está dominada por $|f(x)|$, que está en $L^1(\mathbb{R})$. Así, podemos usar la proposición A.5.

ii) $\forall \omega \in \mathbb{R}$ tenemos $|\hat{f}(\omega)| \leq \int |f(x)| dx = \|f\|_1$. Así, \hat{f} está acotada, y \mathcal{F} es continuo de $L^1(\mathbb{R})$ a $L^\infty(\mathbb{R})$.

iii) Sea $f = \chi_{[a,b]}$ la función característica en $[a, b]$. Tenemos $|\hat{f}(\omega)| \leq 1/(\pi|\omega|)$ para $\omega \neq 0$ tras operar con la integral. Por tanto, el resultado se cumple para cualquier función simple.

Sea ahora una $f \in L^1(\mathbb{R})$; dado que las funciones simples son densas en $L^1(\mathbb{R})$, $\exists (g_n)$ secuencia de funciones simples tal que $\lim_{n \rightarrow \infty} \|f - g_n\|_1 = 0$. Por el resultado anterior,

$$\|\hat{f}(\omega) - \widehat{g_n}(\omega)\|_\infty \leq \|f - g_n\|_1$$

uniformemente en $\omega \in \mathbb{R}$, para n fija. Y por tanto, $\lim_{|\omega| \rightarrow \infty} \hat{f}(\omega) = 0$. ■

La siguiente fórmula es esencial para introducir la transformada inversa de Fourier.

Prop. 7.3- Sean $f, g \in L^1(\mathbb{R})$. Entonces $\hat{f}g, f\hat{g} \in L^1(\mathbb{R})$ y

$$\int f(t)\hat{g}(t)dt = \int \hat{f}(x)g(x)dx. \quad (7.3)$$

Demostración. Por el teorema anterior sabemos que \hat{g} está acotada; y así $f\hat{g} \in L^1(\mathbb{R})$, análogamente para $\hat{f}g$. La igualdad (7.3) viene de utilizar el Teorema de Fubini (A.7). Dado que $e^{-2\pi itx} f(t)g(x) \in L^1(\mathbb{R}^2)$, tenemos

$$\begin{aligned}\int f(t)\hat{g}(t)dt &= \int f(t)\left(\int e^{-2\pi itx}g(x)dx\right)dt = \\ &= \int g(x)\left(\int e^{-2\pi itx}f(t)dt\right)dx = \int \hat{f}(x)g(x)dx \blacksquare\end{aligned}$$

Cabe observar que $\bar{\mathcal{F}}$ tiene estas mismas propiedades, se demuestra sustituyendo i por $-i$.

Una de las ventajas de trabajar con la transformada de Fourier es la facilidad con la que podemos tomar derivadas:

Prop. 7.4- Derivación

i) Si $x^k f(x) \in L^1(\mathbb{R})$, $k = 0, 1, 2, \dots, n$; entonces \hat{f} es diferenciable n veces y

$$\hat{f}^{(k)}(\omega) = \mathcal{F}[(-2\pi ix)^k f(x)](\omega), k = 0, 1, 2, \dots, n. \quad (7.4)$$

ii) Si $f \in C^n(\mathbb{R}) \cap L^1(\mathbb{R})$ y $f^{(k)} \in L^1(\mathbb{R})$ para $k = 1, 2, \dots, n$; entonces

$$\widehat{f^{(k)}}(\omega) = (2\pi i\omega)^k \hat{f}(\omega), k = 0, 1, 2, \dots, n. \quad (7.5)$$

iii) Si $f \in L^1(\mathbb{R})$ tiene soporte acotado, $\hat{f} \in C^\infty(\mathbb{R})$.

Demostración.

i) La función $h: \omega \mapsto e^{-2\pi i\omega x} f(x)$ es infinitamente diferenciable; en concreto, $h^{(k)}(\omega) = (-2\pi ix)^k e^{-2\pi i\omega x} f(x)$ y $|h^{(k)}(\omega)| \leq 2\pi |x^k f(x)|$. Utilizando la proposición A.6 para $k = 1, 2, \dots, n$ nos queda

$$\hat{f}^{(k)}(\omega) = \int e^{-2\pi i\omega x} (-2\pi ix)^k f(x) dx = \mathcal{F}[(-2\pi ix)^k f(x)](\omega).$$

ii) Veámoslo para $n = 1$, el caso general se demuestra análogamente por inducción. Dado que $f \in L^1(\mathbb{R})$, podemos calcular \hat{f}' con

$$\hat{f}'(\omega) = \lim_{a \rightarrow +\infty} \int_{-a}^{+a} e^{-2\pi i\omega x} f'(x) dx.$$

Integrando por partes tenemos

$$\int_{-a}^{+a} e^{-2\pi i\omega x} f'(x) dx = [e^{-2\pi i\omega x} f(x)]_{-a}^{+a} + \int_{-a}^{+a} (2\pi i\omega) e^{-2\pi i\omega x} f(x) dx.$$

Si $\lim_{a \rightarrow \infty} f(\pm a)$ existe, es necesariamente 0, pues f es integrable. Así,

$$\int_{-\infty}^{+\infty} e^{-2\pi i\omega x} f'(x) dx = \int_{-\infty}^{+\infty} (2\pi i\omega) e^{-2\pi i\omega x} f(x) dx.$$

Que es la fórmula para $n = 1$. Nos falta ver que efectivamente los límites existen. Dado que f' es continua y diferenciable tenemos

$$f(a) = f(0) + \int_0^a f'(t) dt;$$

y como $f' \in L^1(\mathbb{R})$, $\lim_{a \rightarrow \infty} \int_0^a f'(t) dt$ existe y el de $f(a)$ también. Análogo para $f(-a)$.

iii) Si $f \in L^1(\mathbb{R})$ tiene soporte acotado ($f(x) = 0 \forall |x|$ mayor que cierta $K > 0$), entonces $x^k f(x)$ es integrable $\forall k \in \mathbb{N}$ y por i) $f \in C^\infty(\mathbb{R})$. \blacksquare

Hemos visto como pasar al dominio de frecuencias, pero nos gustaría encontrar una operación inversa. Como comentamos, resulta que la propia $\bar{\mathcal{F}}$ lo es, pero no siempre se tiene $\hat{f} \in L^1(\mathbb{R})$.

Teo. 7.5- Si $f, \hat{f} \in L^1(\mathbb{R})$, entonces $\bar{\mathcal{F}}(\hat{f}(t)) = f(t)$ en todo punto donde f sea continua.

Demostración. Para cada $n > 0$, introducimos $g_n(x) = e^{-\frac{2\pi}{n}|x|}$, cuya transformada de Fourier es $\widehat{g}_n(\omega) = \frac{1}{\pi} \frac{n}{1+n^2\omega^2}$; y nótese que ambas están en $L^1(\mathbb{R})$. Podemos aplicar la fórmula (7.3) con $f, e^{2\pi itx} g_n(x)$ y tenemos

$$\int_{\mathbb{R}} \hat{f}(x) g_n(x) e^{2\pi itx} dx = \int_{\mathbb{R}} f(u) [e^{2\pi itx} \widehat{g}_n(x)](u) du.$$

Aquí usamos que

$$[e^{2\pi itx} \widehat{g}_n(x)](u) = \int e^{-2\pi iux} e^{2\pi itx} g_n(x) dx = \widehat{g}_n(u - t);$$

y nos queda

$$\int_{\mathbb{R}} \hat{f}(x) g_n(x) e^{2\pi itx} dx = \int_{\mathbb{R}} f(u) \widehat{g}_n(u - t) du.$$

Por otro lado, tenemos $\forall x \in \mathbb{R}, |\hat{f}(x) g_n(x) e^{2\pi itx}| \leq |\hat{f}(x)|$, $\lim_{n \rightarrow \infty} g_n(x) = 1$. Dado que $f \in L^1(\mathbb{R})$ podemos aplicar el Teorema de Lebesgue y pasar el límite dentro de la integral:

$$\lim_{n \rightarrow \infty} \int_{\mathbb{R}} \hat{f}(x) g_n(x) e^{2\pi itx} dx = \int_{\mathbb{R}} \hat{f}(x) e^{2\pi itx} dx = \bar{\mathcal{F}}\hat{f}(t).$$

Por tanto nos falta demostrar que

$$\lim_{n \rightarrow \infty} \int_{\mathbb{R}} f(u) \widehat{g}_n(u - t) du = f(t).$$

Veamos que dado que $\widehat{g}_n \in L^1(\mathbb{R})$,

$$\int_{\mathbb{R}} \widehat{g}_n(\omega) d\omega = \lim_{a \rightarrow \infty} \int_{-a}^{+a} \frac{1}{\pi} \frac{n}{1+n^2\omega^2} d\omega = 1;$$

$$\int_{\mathbb{R}} f(u) \widehat{g}_n(u - t) du - f(t) = \int_{\mathbb{R}} (f(u + t) - f(t)) \widehat{g}_n(u) du.$$

Dado $\epsilon > 0, \exists \delta > 0$ tq. $|y - t| \leq \delta \Rightarrow |f(y) - f(t)| \leq \epsilon$. Descomponemos la integral

$$\begin{aligned} & \int_{\mathbb{R}} (f(u + t) - f(t)) \widehat{g}_n(u) du = \\ &= \int_{|u| \leq \delta} (f(u + t) - f(t)) \widehat{g}_n(u) du + \int_{|u| > \delta} (f(t + u) - f(t)) \widehat{g}_n(u) du. \end{aligned}$$

Para todo $n > 0$,

$$\int_{|u| \leq \delta} (f(t + u) - f(t)) \widehat{g}_n(u) du \leq \epsilon \int_{|u| \leq \delta} |\widehat{g}_n(u)| du \leq \epsilon;$$

y nos falta sólo ver

$$\lim_{n \rightarrow \infty} \int_{|u| > \delta} (f(t + u) - f(t)) \widehat{g}_n(u) du = 0.$$

Para ello tenemos

$$\left| f(t) \int_{|u| > \delta} \widehat{g}_n(u) du \right| = |f(t)| \left(1 - \frac{2}{\pi} \arctan(n\delta) \right).$$

Y dado que \widehat{g}_n es par y decreciente en \mathbb{R}_+ ,

$$\left| \int_{|u| > \delta} f(t + u) \widehat{g}_n(u) du \right| \leq \widehat{g}_n(\delta) \|f\|_1.$$

El lado derecho de las últimas dos expresiones tiende a 0; hemos terminado. ■

La siguiente proposición nos da condiciones suficientes para que \hat{f} sea integrable.

Prop. 7.6- Si $f \in C^2(\mathbb{R})$ y $f, f', f'' \in L^1(\mathbb{R})$, entonces $\hat{f} \in L^1(\mathbb{R})$.

Demostración. Por la propiedad de la derivación (7.5), tenemos $\widehat{f''}(\omega) = -4\pi^2 \omega^2 \hat{f}(\omega)$. Por otro lado, por el teorema 7.2 tenemos $\lim_{|\omega| \rightarrow \infty} |\widehat{f''}(\omega)| = 0$.

Así, $\exists M > 0$ tq. $\forall |\omega| > M, 4\pi^2 |\omega|^2 |\hat{f}(\omega)| \leq 1$. Al ser $|\hat{f}(\omega)|$ continuo en \mathbb{R} y menor a $1/(4\pi^2 \omega^2)$ en el infinito, se tiene que $\hat{f} \in L^1(\mathbb{R})$. ■

Estos resultados nos indican que es necesario restringir la elección de funciones de $L^1(\mathbb{R})$ si queremos que la inversa esté bien definida. Para ello introduciremos un nuevo subespacio cerrado por las distintas operaciones.

Def. 7.7- Una función $f: \mathbb{R} \rightarrow \mathbb{C}$ se dice que es de decrecimiento rápido si $\forall p \in \mathbb{N}, \lim_{|x| \rightarrow \infty} |x^p f(x)| = 0$.

Prop. 7.8- Si f es localmente integrable y decrece rápidamente, entonces $\forall p \in \mathbb{N}, x^p f(x) \in L^1(\mathbb{R})$.

Demostración. Dado que f decae rápidamente,

$$\exists M > 0 \text{ tq. } \forall |x| > M, |x^{p+2} f(x)| \leq 1.$$

Así,

$$\begin{aligned} \int_{\mathbb{R}} |x^p f(x)| dx &\leq \int_{|x| \leq M} |x^p f(x)| dx + \int_{|x| > M} \frac{1}{x^2} |x^{p+2} f(x)| dx \leq \\ &\leq M^p \int_{|x| \leq M} |f(x)| dx + \int_{|x| > M} \frac{1}{x^2} dx < +\infty \quad \blacksquare \end{aligned}$$

Prop. 7.9- Si $f \in L^1(\mathbb{R})$ decae rápidamente, entonces \hat{f} es infinitamente diferenciable.

Demostración. Esto es: $\forall p \in \mathbb{N}, x^p f(x) \in L^1(\mathbb{R})$ por el resultado anterior, ya que f decae rápidamente. Si ahora usamos la Prop. 7.4 tenemos $\hat{f} \in C^\infty$ ■

Def. 7.10- El espacio de Schwartz, o $S(\mathbb{R})$, es el espacio funciones $f: \mathbb{R} \rightarrow \mathbb{C}$ tales que
i) f es infinitamente diferenciable.
ii) f y todas sus derivadas son de decrecimiento rápido.

Prop. 7.11- El espacio $S(\mathbb{R})$ tiene las siguientes propiedades:
i) S es invariante bajo la multiplicación por un polinomio.
ii) S es invariante bajo la derivación.
iii) $S \subset L^1(\mathbb{R})$.

Teo. 7.12- El espacio $S(\mathbb{R})$ es invariante bajo la transformada de Fourier; esto es, $f \in S(\mathbb{R}) \Rightarrow \hat{f} \in S(\mathbb{R})$.

Demostración. Si $f \in S(\mathbb{R})$, entonces f está en $L^1(\mathbb{R})$ y decae rápidamente. Por la proposición 7.9, $\hat{f} \in C^\infty$, la primera condición que buscamos comprobar.

Dado que $\forall k$ $f^{(k)}$ decrece rápidamente, \hat{f} es también integrable para cualquier k por la proposición 7.8. Estos hechos nos dicen que \hat{f} decrece rápidamente, pues por las reglas de derivación:

$$\forall k \in \mathbb{N}, \widehat{f^{(k)}}(\omega) = (2\pi i \omega)^k \hat{f}(\omega);$$

y por el teorema 7.2, $\lim_{|\omega| \rightarrow +\infty} |\omega|^k |\hat{f}(\omega)| = 0$.

Queremos ver que todas las derivadas de \hat{f} decrecen rápidamente. Dado que $((-2\pi i x)^q f(x))^{(p)}$ es integrable, de nuevo por la proposición 7.4 vemos

$$\frac{1}{(2\pi i)^p} \mathcal{F} \left(((-2\pi i x)^q f(x))^{(p)} \right) (\omega) = \omega^p \mathcal{F}((-2\pi i x)^q f(x))(\omega) = \omega^p \hat{f}^{(q)}(\omega).$$

El lado derecho es por tanto la transformada de Fourier de una función integrable, y por el Teorema de Riemann-Lebesgue $\lim_{|\omega| \rightarrow \infty} |\omega^p \hat{f}^{(q)}(\omega)| = 0$. ■

Vimos en el teorema 7.5 que si f y \hat{f} son integrables, en todo punto donde f sea continua

$$f(x) = \int_{\mathbb{R}} e^{2\pi i x \omega} \hat{f}(\omega) d\omega.$$

Si $f \in S(\mathbb{R})$, entonces \hat{f} también. Dado que f es continua, lo anterior es cierto $\forall x \in \mathbb{R}$. Dicho de otra forma,

$$f = \bar{\mathcal{F}} (\mathcal{F} f) = \mathcal{F} (\bar{\mathcal{F}} f) \quad \forall f \in S(\mathbb{R}).$$

Teo. 7.13- La transformada de Fourier \mathcal{F} es una transformación lineal biyectiva de $S(\mathbb{R})$ a sí mismo. La operación inversa es $\mathcal{F}^{-1} = \bar{\mathcal{F}}$; se tiene

$$\hat{f}(\omega) = \int_{\mathbb{R}} e^{-2\pi i \omega x} f(x) dx,$$

$$f(x) = \int_{\mathbb{R}} e^{+2\pi i x \omega} \hat{f}(\omega) d\omega;$$

para cualquier $f \in S(\mathbb{R})$ y todo $x, \omega \in \mathbb{R}$.

Para cerrar esta sección introduciremos un concepto al que volveremos en el futuro: la convolución de funciones.

Def. 7.14- La convolución de dos funciones $f, g: \mathbb{R} \rightarrow \mathbb{C}$ es

$$f * g(x) = \int_{\mathbb{R}} f(x-t)g(t)dt = \int_{\mathbb{R}} f(u)g(x-u)du,$$

si existe.

Nótese que en general esta integral no tiene por qué existir. Podemos restringir las funciones que usamos para asegurar su existencia.

Prop. 7.15- Si $f, g \in L^1(\mathbb{R})$ se tiene que $f * g \in L^1(\mathbb{R})$ y está definida casi por todo. Además, se trata de un operador bilineal con

$$\|f * g\|_1 \leq \|f\|_1 \|g\|_1.$$

Demostración. Dado que $f, g \in L^1(\mathbb{R})$, podemos usar el Teorema de Fubini para ver que la función $(y, z) \mapsto f(y)g(z) \in L^1(\mathbb{R}^2)$. Con el cambio de variables $y = x - t, z = t$:

$$\int_{\mathbb{R}} \int_{\mathbb{R}} f(y)g(z)dydz = \int_{\mathbb{R}} \int_{\mathbb{R}} f(x - t)g(t)dxdt;$$

y por tanto la función $x \mapsto \int_{\mathbb{R}} f(x - t)g(t)dxdt$ está definida casi por todo y pertenece a $L^1(\mathbb{R})$ por el mismo teorema.

Para la desigualdad, tenemos

$$|f * g(x)| \leq \int_{\mathbb{R}} |f(x - t)||g(t)|dt = |f| * |g|(x),$$

y

$$\begin{aligned} \int_{\mathbb{R}} |f * g(x)|dx &\leq \int_{\mathbb{R}} |f| * |g|(x)dx = \int_{\mathbb{R}} dx \int_{\mathbb{R}} |f(x - t)||g(t)|dt = \\ &= \int_{\mathbb{R}} |g(t)| \left(\int_{\mathbb{R}} |f(x - t)|dx \right) dt = \|g\|_1 \|f\|_1 \quad \blacksquare \end{aligned}$$

De hecho, se puede demostrar que

$$f, g \in S(\mathbb{R}) \Rightarrow f * g \in S(\mathbb{R}).$$

El resultado clave para nosotros será el siguiente:

Prop. 7.16- Si $f, g \in S(\mathbb{R})$, se tiene

- i) $\widehat{f * g} = \hat{f} \cdot \hat{g}.$
- ii) $\widehat{f \cdot g} = \hat{f} * \hat{g}.$

Demostración. Sabemos que $f * g \in L^1(\mathbb{R})$, podemos calcular su transformada:

$$\begin{aligned} \int_{\mathbb{R}} e^{-2\pi i \omega x} (f * g(x))dx &= \int_{\mathbb{R}} e^{-2\pi i \omega x} \left(\int_{\mathbb{R}} f(x - t)g(t)dt \right) dx = \\ &= \int_{\mathbb{R}} g(t) \left(\int_{\mathbb{R}} f(x - t)e^{-2\pi i \omega x} dx \right) dt = \int_{\mathbb{R}} g(t)e^{-2\pi i \omega t} \hat{f}(\omega)dt = \hat{g}(\omega) \cdot \hat{f}(\omega) \end{aligned}$$

Para la segunda igualdad, nótese que estos cálculos son también válidos para $\bar{\mathcal{F}}$ cambiando i por $-i$. Aplicando i) sobre \hat{f}, \hat{g} tenemos

$$\bar{\mathcal{F}}(\hat{f} * \hat{g})(x) = \bar{\mathcal{F}}\hat{f}(x) \cdot \bar{\mathcal{F}}\hat{g}(x) = f(x) \cdot g(x) \quad \blacksquare$$

Para cerrar esta sección, merece la pena asegurarnos de que comprendemos lo que estos resultados representan. La transformada de Fourier de una señal puede entenderse como una extensión de la serie de Fourier cuando el periodo (a) tiende a infinito. Obtenemos el espectro de la función, que nos indica qué amplitud corresponde a cada frecuencia que la compone, así como su fase.

Llegados a este punto, tenemos herramientas suficientemente potentes como para ir y venir del espacio temporal al de frecuencias (siempre que la función cumpla los requisitos). Además, con este último teorema podemos calcular qué efectos tendrán en la señal modificaciones en el espectro, y viceversa. Se trata de un resultado clave para entender el funcionamiento de la transformada.

8. Teorema de Shannon

Adentrémonos aún más en el aspecto práctico de este tema estudiando el proceso de muestreo de una señal. En las aplicaciones trabajaremos generalmente sólo con muestras, y necesitamos conocer el trasfondo teórico.

Hay dos pasos relevantes aquí: *muestreo* y *cuantificación*.

Def. 8.1- Cuantificación es el proceso de asignar valores numéricos a las muestras y requiere la definición de una escala, entre otras cosas.

La base de este paso es de carácter computacional, por lo que no lo trataremos aquí. Basta mencionar que se trata de una pérdida inevitable de información, pues estamos redondeando las medidas a unos ciertos valores. Dicho esto, una cuantización de buena calidad hace que esta pérdida sea imperceptible.

Def. 8.2- El muestreo consiste en la toma de muestras de la señal a intervalos regulares. La frecuencia con la que se toman las muestras es la llamada *frecuencia de muestreo*, medida en Hz.

Al tomar muestras, hay dos ideas intuitivas: que se produce siempre una cierta pérdida de información y que cuanta más muestras tomemos, menor es esta pérdida. Veremos que lo primero no es necesariamente cierto, y lo segundo aunque cierto no resulta práctico. No es factible aumentar el número de muestras indefinidamente, tanto por razones físicas como económicas o prácticas. Nos centraremos en el carácter matemático de este paso y, en concreto, veremos el Teorema de Shannon, el resultado más importante en este ámbito.

Matemáticamente, el proceso de muestreo equivale a multiplicar la función por el *peine de Dirac*: una distribución³ definida por

$$\Delta_a = \sum_{n=-\infty}^{+\infty} \delta_{na}.$$

δ_{na} es la llamada *delta de Dirac*, la distribución que aplicada a una función devuelve su valor en na . Así, si f es una función, su producto con el peine es la distribución

$$f\Delta_a = \sum_{n=-\infty}^{+\infty} f(na)\delta_{na}.$$

Para ser más precisos, $af\Delta_a$ es la distribución que “aproxima” f , pues

$$f \approx \sum_{n=-\infty}^{+\infty} f(na)\chi_{[na-\frac{a}{2}, na+\frac{a}{2})} \approx \sum_{n=-\infty}^{+\infty} f(na)a\delta_{na}; \text{ o } f \approx af\Delta_a.$$

Luego volveremos a esta idea, primero debemos ver un resultado clave.

³ Las distribuciones son un tema extenso al que lamentablemente no podemos entrar aquí. Veremos los resultados correspondientes de forma superficial, tan sólo en el aspecto que nos concierna. Para una introducción a las distribuciones y sus usos en la teoría de Fourier, véase [6].

Teo. 8.3- Fórmula de Poisson: La ecuación

$$\sum_{n=-\infty}^{+\infty} f(t - na) = \frac{1}{a} \sum_{n=-\infty}^{+\infty} \hat{f}\left(\frac{n}{a}\right) e^{2\pi i n \frac{t}{a}}, \quad (8.6)$$

donde $a > 0$ es arbitrario y fijo, con su versión dual

$$\sum_{n=-\infty}^{+\infty} \hat{g}\left(\lambda - \frac{n}{a}\right) = a \sum_{n=-\infty}^{+\infty} g(na) e^{-2\pi i n \lambda a}, \quad (8.7)$$

es la llamada fórmula de Poisson.

Nótese que el lado derecho de (8.6) no está definido en general, es necesario que \hat{f} sea continua para que pueda ser evaluada en cualquier $\left(\frac{n}{a}\right)$. Para el lado izquierdo nos basta con que la serie converja. Nos centraremos más tarde en el caso $f \in S(\mathbb{R})$, donde sabemos que \hat{f} se comporta bien.

Para llegar a la demostración veamos algunos resultados previos:

Lema 8.4- Si $f \in L^1(\mathbb{R})$, $a > 0$, la serie

$$F(t) = \sum_{n=-\infty}^{+\infty} f(t - na);$$

i) converge en $L^1(0, a)$, y $F \in L_p^1(0, a)$. Sus coeficientes de Fourier son

$$c_k(F) = \frac{1}{a} \hat{f}\left(\frac{k}{a}\right), \quad k \in \mathbb{Z}.$$

ii) Si además $f' \in L^1(\mathbb{R})$, $F(t)$ converge uniformemente en \mathbb{R} , y por tanto es continua.

Demostración.

i) Veamos primero que la serie converge en $(0, a)$ comprobando que

$$F_N(t) = \sum_{n=-N}^N f(t - na)$$

es una secuencia de Cauchy. Tenemos

$$\|F_{N+P} - F_N\|_{L^1(0, a)} \leq \sum_{N < |n| \leq N+P} \int_0^a |f(t - na)| dt \leq \int_{|x| \geq Na} |f(x)| dx.$$

Dado que $f \in L^1(\mathbb{R})$, esta integral tiende a 0 si $N, P \rightarrow \infty$. Por tanto, F_N converge a F en $(0, a)$ y $F \in L_p^1(0, a)$ por periodicidad. Los coeficientes de Fourier de F son el límite de los de F_N :

$$c_k(F_N) = \frac{1}{a} \sum_{|n| \leq N} \int_0^a f(t - na) e^{-2\pi i k t / a} dt = \frac{1}{a} \int_{-Na}^{(N+1)a} f(x) e^{-2\pi i k x / a} dx;$$

$$c_k(F) = \lim_{N \rightarrow \infty} c_k(F_N) = \frac{1}{a} \int_{-\infty}^{+\infty} f(x) e^{-2\pi i k x / a} dx = \frac{1}{a} \hat{f}\left(\frac{k}{a}\right);$$

como queríamos demostrar.

ii) Sea $a = 1$ para simplificar notación. Veremos que existe una secuencia de números positivos u_n tal que

$$|f(t - n)| \leq u_n \quad \forall t \in [0, 1], n \in \mathbb{Z};$$

y tal que $\sum_{n=-\infty}^{+\infty} u_n < +\infty$.

Consideremos las siguientes igualdades:

$$\begin{aligned} f(t-n) - \int_n^{n+1} f(t-x)dx &= \int_n^{n+1} (f(t-n) - f(t-x))dx, \\ f(t-n) - f(t-x) &= \int_n^x f'(t-y)dy. \end{aligned}$$

Con ambas relaciones:

$$\left| f(t-n) - \int_n^{n+1} f(t-x)dx \right| \leq \int_n^{n+1} |f'(t-y)|dy;$$

esto implica que:

$$|f(t-n)| \leq \int_n^{n+1} (|f(t-x)| + |f'(t-x)|)dx \leq \int_{-(n-1)}^{-(n+1)} (|f(u)| + |f'(u)|)du.$$

Sean ahora

$$\begin{aligned} u_n &= \int_{-(n-1)}^{-(n+1)} (|f(u)| + |f'(u)|)du; \\ \sum_{n=-\infty}^{+\infty} u_n &= 2 \int_{\mathbb{R}} (|f(u)| + |f'(u)|)du < +\infty. \end{aligned}$$

Esto demuestra que $F(t)$ converge uniformemente en $[0,1]$, $[0,a]$ en general. Dado que F es periódica, nos queda que la convergencia es uniforme en todo \mathbb{R} ■

Teo. 8.5- Si $f, f' \in L^1(\mathbb{R})$, la fórmula de Poisson (8.6) se cumple para todo $t \in \mathbb{R}$. En concreto, el lado izquierdo converge en \mathbb{R} a una función F y el lado derecho, su serie de Fourier, converge uniformemente en \mathbb{R} a F .

Demostración. Veamos que F es de variación acotada, y así junto al lema anterior podremos aplicar el teorema 4.6. Por simplicidad tomamos de nuevo $a = 1$.

Sea $0 = t_0 < t_1 \dots < t_p < t_{p+1} = 1$ una subdivisión de $[0,1]$. Entonces

$$\begin{aligned} M_p &= \sum_{i=0}^p |F(t_{i+1}) - F(t_i)| \leq \sum_{i=0}^p \sum_{n=-\infty}^{+\infty} |f(t_{i+1}-n) - f(t_i-n)| = \\ &= \sum_{n=-\infty}^{+\infty} \sum_{i=0}^p |f(t_{i+1}-n) - f(t_i-n)| \leq \sum_{n=-\infty}^{+\infty} \sum_{i=0}^p \int_{t_i}^{t_{i+1}} |f'(t-n)|dt. \end{aligned}$$

Finalmente,

$$M_p \leq \sum_{n=-\infty}^{+\infty} \int_0^1 |f'(t-n)|dt = \sum_{n=-\infty}^{+\infty} \int_{-n}^{-(n-1)} |f'(u)|du = \int_{\mathbb{R}} |f'(u)|du.$$

M_p está acotado independientemente de la subdivisión, y por tanto F es de variación acotada. ■

Con la fórmula de Poisson demostrada, pasemos a ver el tema central de esta sección. Teníamos que el proceso de muestreo cada a segundos equivale a

$$af\Delta_a = a \sum_{n=-\infty}^{+\infty} f(na)\delta_{na}.$$

Empecemos estudiando qué forma tiene el espectro de $af\Delta_a$. Por la proposición 7.16⁴,

$$f \cdot \Delta_a = \bar{\mathcal{F}}(\hat{f} * \hat{\Delta}_a),$$

y si ahora tomamos la transformada de ambos lados:

$$\widehat{f \cdot \Delta_a} = \hat{f} * \hat{\Delta}_a.$$

Para desarrollar la expresión buscaremos $\hat{\Delta}_a$. Dado que se trata de una distribución, no entraremos aquí al cálculo (podemos encontrarlo en [6]), pero nos basta con saber que obtenemos

$$\begin{aligned}\hat{\Delta}_a &= \Delta_{\frac{1}{a}}; \\ a\widehat{f \cdot \Delta_a} &= a\hat{f} * \Delta_{\frac{1}{a}}.\end{aligned}$$

El lado derecho de esta igualdad se puede calcular directamente, y utilizando (8.7) llegamos a un resultado clave:

$$a\widehat{f \cdot \Delta_a}(\lambda) = a\hat{f} * \Delta_{\frac{1}{a}}(\lambda) = a \sum_{n=-\infty}^{+\infty} \hat{f}\left(\lambda - \frac{n}{a}\right) = a \sum_{n=-\infty}^{+\infty} f(na)e^{-2\pi ina\lambda}.$$

De aquí deducimos que el espectro del muestreo tiene **periodo $1/a$** . En concreto, este espectro consiste en la suma de copias del espectro de f trasladadas n/a unidades, $n \in \mathbb{Z}$. Introducimos ahora un nuevo conjunto de funciones:

Def. 8.6- Una función f se dice que está limitada en banda si existe un número λ_c tal que f no contiene frecuencias mayores que λ_c .

O lo que es lo mismo,

$$\text{supp}(\hat{f}) \subset [-\lambda_c, \lambda_c].$$

En este conjunto encontramos un resultado clave. Estudiemos qué ocurre al tomar muestras de una función limitada en banda; se nos presentan dos casos:

- i) $\frac{1}{a} < \lambda_c$. Esto implica que en el espectro de la función muestreada las traslaciones del espectro original están solapadas.
- ii) $\frac{1}{a} \geq \lambda_c$. En el espectro de la función muestreada no tenemos solapamiento y cada copia trasladada está aislada.

En la figura 1 podemos ver mucho más claramente todas estas ideas. La primera pareja corresponde a una función $f(t)$ limitada en banda cualquiera y su transformada. El espectro es nulo fuera de $[-\lambda_c, \lambda_c]$ por definición.

La segunda tiene primero a la función muestreada, con valores en los múltiplos de a . Su transformada es, como decíamos, un conjunto de copias centradas en los múltiplos de $1/a$. En este ejemplo concreto tenemos $1/a \geq \lambda_c$, pero vemos que si no fuera así las copias del espectro estarían solapadas. En ese caso, no podríamos saber el valor de una sola copia donde se produjera este solapamiento.

⁴ Nótese que este resultado lo demostramos para funciones y no distribuciones, pero en efecto se puede extender.

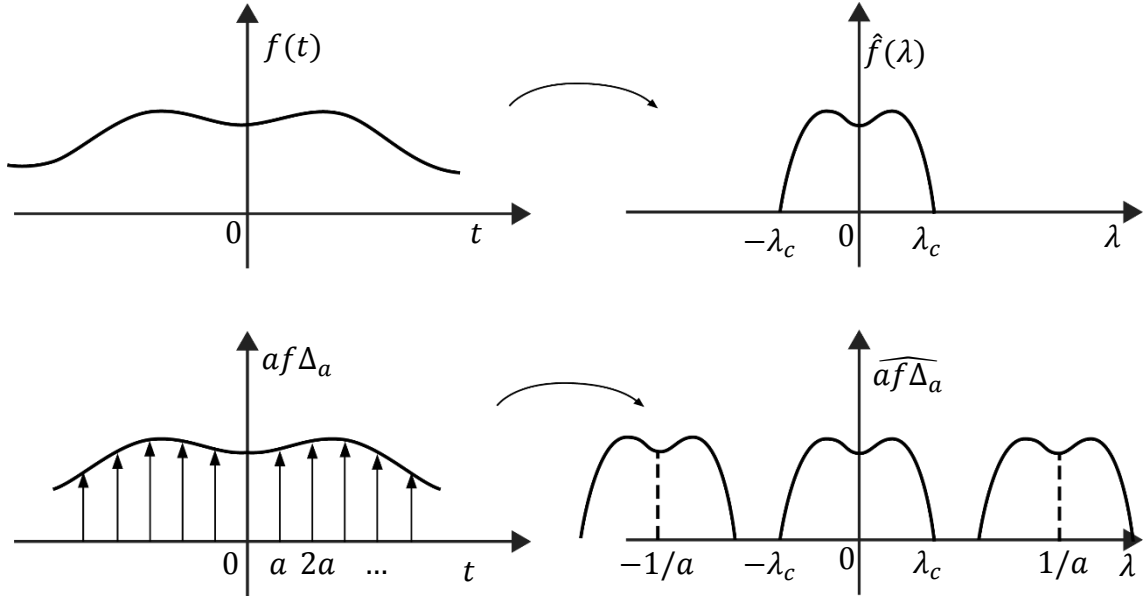


Figura 1: En la fila superior: la función original y su transformada. En la inferior, la pareja análoga para la función muestreada.

Con estas definiciones podemos enunciar el Teorema.

Teo. 8.7- (Teorema de Muestreo de Shannon) Sea f función limitada en banda, sin frecuencias de más de $\lambda_c \text{Hz}$ y tal que $\hat{f} \in L^2(\mathbb{R})$. Entonces

$$\forall a > 0, \sum_{n=-\infty}^{+\infty} |f(na)|^2 < +\infty,$$

y si $1/a \geq 2\lambda_c$,

$$f(t) = \sum_{n=-\infty}^{\infty} f(na) \frac{\sin\left(\frac{\pi}{a}(t - na)\right)}{\frac{\pi}{a}(t - na)}$$

en $L^2(\mathbb{R})$. Si además

$$\sum_{n=-\infty}^{+\infty} |f(na)| < +\infty,$$

la serie converge uniformemente en \mathbb{R} y la igualdad es cierta $\forall t \in \mathbb{R}$.

Demostración. La idea detrás del Teorema de Shannon es tomar la copia central aislada y recrear la función original. Tenemos, por *Poisson*,

$$\sum_{n=-\infty}^{+\infty} \hat{f}\left(\lambda - \frac{n}{a}\right) = a \sum_{n=-\infty}^{+\infty} f(na) e^{-2\pi i n \lambda a}.$$

El lado izquierdo es una función periódica e integrable en uno de sus periodos. Así, podemos expresarlo con una serie de Fourier:

$$\sum_{n=-\infty}^{+\infty} \hat{f}\left(\lambda - \frac{n}{a}\right) = F(\lambda) = \sum_{n=-\infty}^{+\infty} c_n e^{2\pi i n \lambda a}.$$

Se trata de una igualdad en $L^2(0, 1/a)$, y dado que los coeficientes son únicos:

$$c_{-n} = af(na);$$

y nos queda, por Parseval,

$$\forall a > 0, \sum_{n=-\infty}^{+\infty} |f(na)|^2 < +\infty.$$

Consideremos ahora la función característica $r(\lambda) = \chi_{[-\frac{1}{2a}, \frac{1}{2a}]}(\lambda)$. Si la multiplicamos por $F(\lambda)$, estamos aislando la copia central, y por tanto tenemos que en $L^2(\mathbb{R})$

$$\hat{f}(\lambda) = a \sum_{n=-\infty}^{+\infty} f(na)r(\lambda)e^{-2\pi in\lambda a}.$$

Usando la continuidad de $\bar{\mathcal{F}}$ en $L^2(\mathbb{R})$ y que

$$\bar{\mathcal{F}}[r(\lambda)e^{-2\pi in\lambda a}] = (\bar{\mathcal{F}} r)(t - na) = \frac{\sin\left(\frac{\pi}{a}(t - na)\right)}{\pi(t - na)},$$

obtenemos finalmente

$$f(t) = \sum_{n=-\infty}^{+\infty} f(na) \frac{\sin\left(\frac{\pi}{a}(t - na)\right)}{\frac{\pi}{a}(t - na)},$$

con el lado derecho convergente en $L^2(\mathbb{R})$. Pero si además

$$\sum_{n=-\infty}^{+\infty} |f(na)| < +\infty,$$

tendremos que la convergencia es uniforme en \mathbb{R} a una función g . Esto implica que converge a g en $L^2(J)$, donde J es cualquier intervalo acotado. Así, $f = g$ casi por todo en \mathbb{R} , y al ser continuas nos queda $f(t) = g(t) \forall t \in \mathbb{R}$ ■

La importancia de este teorema es evidente: si la señal original está limitada en banda, podemos muestrearla sin pérdida alguna de información. No sólo eso, sino que además nos da la frecuencia de muestreo suficiente para ello.

Como decíamos previamente, si la frecuencia de muestreo no es suficientemente alta tendremos un solapamiento entre las diversas copias del espectro de la función muestreada. En ese caso no se podrá reconstruir correctamente la señal original y se produce el llamado *aliasing*.

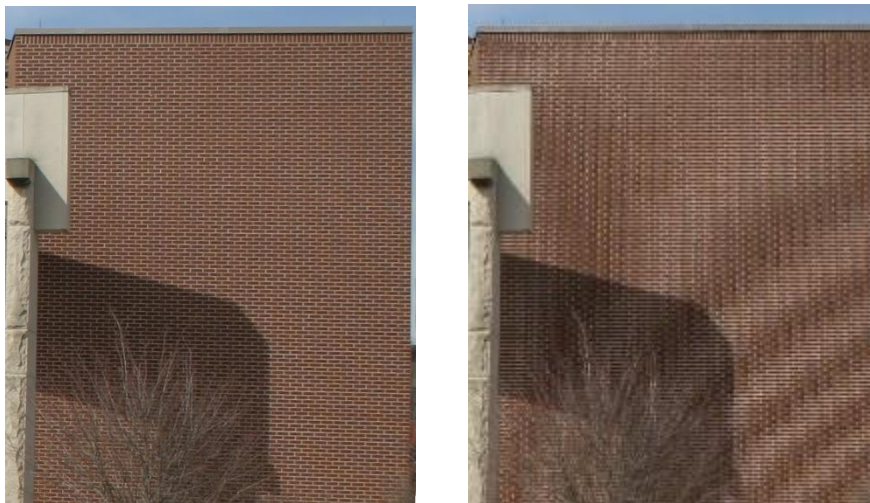


Figura 2: Imagen sin y con aliasing.

En la figura 2 vemos un caso de *aliasing* espacial muy evidente: en la izquierda la imagen sin defectos, en la derecha deformada. La pared presenta un patrón de frecuencia espacial alta: la forma de los ladrillos se repite en poco espacio. Así, tomar una foto suya con poca resolución (con baja frecuencia de muestreo) provocará las ondas que vemos en la foto de la derecha. Este fenómeno explica por qué, por ejemplo, los reporteros de televisión no pueden llevar ropa de cuadros o rayas; o por qué el fondo de imágenes puede verse borroso.

El teorema de Shannon y sus consecuencias están presentes en casi cualquier campo donde sea necesario muestrear. Los más simples son en sonido e imagen, pero también en telecomunicaciones y todo tipo de procesos de digitalización. Aunque pueda parecer que el necesitar funciones limitadas en banda sea muy restrictivo, en la práctica este conjunto cubre un amplio abanico de casos. En comunicaciones (señal AM, FM, etc.), las señales suelen tener asignadas unos ciertos rangos de frecuencias, con lo que la aplicación del Teorema es evidente.

En muchos otros casos, la presencia de frecuencias altas es inapreciable, y podemos reconstruir la señal casi a la perfección. De hecho en la práctica se suele filtrar la señal para eliminar frecuencias a partir de cierto punto previamente, y así evitar problemas. Dicho todo esto, no es fácil juzgar a priori cuál será la frecuencia máxima de una función limitada en banda y cada caso puede variar.

Un ejemplo esclarecedor de estas ideas es el proceso de grabar audio. Los sonidos que percibimos no son más que vibraciones (en el aire) de ciertas frecuencias; cuanto mayor sea la frecuencia, más agudo el sonido. Estas vibraciones pueden ser modeladas por una señal que representa la presión en el aire en función del tiempo y, gracias a la transformada de Fourier, podemos descomponerla.

El hecho clave es que los humanos sólo podemos percibir sonidos de frecuencias entre 20Hz y 20000Hz . Sonidos de más o menos frecuencia son completamente inaudibles para nosotros. Así, si grabamos sonido con una frecuencia de muestreo mayor al doble de 20000Hz (la más común es 44100Hz) podremos reconstruir el sonido sin problemas.

Para cerrar esta sección, veamos cómo usar la FFT para calcular una aproximación del espectro, pues en muchas ocasiones querremos calcular la transformada a partir de unas muestras. Supongamos la señal f limitada en banda, $\text{supp}(\hat{f}) \subset [-\lambda_c, \lambda_c]$.

Tenemos que

$$\frac{1}{a} \geq 2\lambda_c, \quad \hat{f}(\lambda) = a \sum_{n=-\infty}^{+\infty} f(na) e^{-2\pi i n a \lambda} \quad \forall \lambda \in [-\lambda_c, \lambda_c].$$

Supongamos que la señal se mide durante $t \in [-Na, (N-1)a]$. La aproximación del espectro, basada en las muestras $x_n = f(na)$, $n = -N, \dots, N-1$ será:

$$S_N(\lambda) = a \sum_{n=-N}^{N-1} x_n e^{-2\pi i n a \lambda},$$

y sus valores en los puntos $\lambda_k = k/(2Na) = k/T$,

$$S_N\left(\frac{k}{T}\right) = a \sum_{n=-N}^{N-1} x_n e^{-\pi i n / N},$$

pueden ser calculados usando la FFT como se describe en la sección 6.

9. Filtros Digitales

A lo largo de esta sección estudiaremos los filtros digitales: sistemas que realizan operaciones sobre una señal discreta, con ciertas condiciones. Operando y combinando valores de la original, podemos formar una señal nueva que busquemos por diversas razones. Consideremos un ejemplo muy simple:

$$z_n = \frac{1}{4}(x_{n-1} + 2x_n + x_{n+1}); n = 0, 1, \dots, N-1.$$

Este filtro transformará una señal (su conjunto de muestras) x_n en una nueva, z_n . En concreto, busca reducir la importancia de altas frecuencias tomando por cada punto una media ponderada de los valores de la original. No entraremos aquí en detalle sobre los coeficientes, pero la idea queda clara.

Recordemos que la señal original es periódica, y por tanto la de salida también lo es. Se trata de una transformación lineal entre vectores de tamaño N , y podemos pues expresar el filtro en forma matricial, de forma que $z = Sx$. En este caso:

$$S = \frac{1}{4} \begin{pmatrix} 2 & 1 & 0 & & 0 & 1 \\ 1 & 2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 2 & & 0 & 0 \\ 0 & 0 & 1 & & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 1 & 0 & 0 & & 1 & 2 \end{pmatrix}.$$

Nótese cómo tenemos 1 s en las esquinas superior derecha e inferior izquierda. Esto se debe a que se trata de una señal periódica, y por tanto a la hora de hacer cálculos el vecino izquierdo de x_0 es " x_{-1} " = x_{N-1} . La forma de la matriz es relevante, veamos la definición formal del conjunto de matrices con esta estructura:

Def. 9.1- Una matriz $N \times N$ es de Toeplitz si sus elementos son constantes a lo largo de cada diagonal. Una matriz es *circulante* si además de ser Toeplitz cumple

$$S_{(k+s) \bmod N, (l+s) \bmod N} = S_{k,l} \quad \forall k, l \in [0, N-1]; \forall s \in N.$$

donde S es la matriz.

Coloquialmente, esto significa que cada fila es la anterior desplazada un elemento a la derecha. De esta definición es claro que una matriz circulante viene definida unívocamente por sus elementos en la primera columna. Así, dado un filtro de la forma

$$z_n = \sum_k t_k x_{n-k}, k \in \mathbb{Z},$$

(suponemos $k \in [-N/2, N/2)$ reagrupando gracias a la periodicidad) podemos construir la matriz correspondiente usando como primera columna el vector s definido por:

$$s_l = \begin{cases} t_l, & \text{si } 0 \leq l < N/2 \\ t_{l-N}, & \text{si } \frac{N}{2} \leq l \leq N-1 \end{cases}.$$

Como es una matriz de Toeplitz circulante, es directo construir el resto. El proceso inverso es también posible, para pasar de la matriz al filtro explícito. Hasta ahora todo esto no está relacionado con el análisis de Fourier, pasemos a ello ahora. Definiremos matemáticamente que se considera un filtro digital y veremos algunos resultados.

Utilizaremos la notación de la sección de la DFT, Transformada Discreta de Fourier recordemos:

- i. ϕ_n ; los tonos digitales puros que conforman la base de Fourier (definición 5.2).
- ii. F_N ; la matriz de cambio a la base de Fourier de la DFT (definición 5.4).
- iii. F_N^H ; la matriz adjunta de F_N y correspondiente a la operación inversa.

Def. 9.2- Una transformación lineal $S: R^N \rightarrow R^N$ es un filtro digital si envía vectores de la base de Fourier en R^N a múltiplos de sí mismo, esto es,

$$\forall n \in [0, N-1], \exists \lambda_{S,n} \text{ tq. } S(\phi_n) = \lambda_{S,n} \phi_n.$$

Es decir, los N vectores de Fourier son los vectores propios de S . El vector de valores propios de S , $\lambda_S = (\lambda_{S,n})_{n=0}^{N-1}$; es la llamada respuesta en frecuencia de S .

Por la definición de F_N , un filtro S se puede expresar como

$$S = F_N^H D F_N;$$

ya que las columnas de F_N^H son estos vectores. Aquí, D es la matriz de respuesta en frecuencias: su diagonal es el vector de valores propios.

Corolario 9.3- Todos los filtros digitales conmutan, esto es, si S_1 y S_2 son filtros:

$$S_1 S_2 = S_2 S_1.$$

Conviene recalcar qué significa exactamente esta definición. De la descomposición de la matriz vemos que multiplicarla por un vector equivale a tres pasos: cambiar a la base de Fourier, modificar los coeficientes (de cada frecuencia) como indique la matriz D y finalmente deshacer la transformada. Esto es, un filtro no es más que una modificación concreta en el plano de frecuencias de la señal.

Ahora podemos conectar estas nociones con lo tratado a comienzos de este capítulo:

Teo. 9.4- Una transformación lineal S es un filtro digital si y sólo si es una matriz circulante de Toeplitz.

Demostración. Que sea un filtro es equivalente a si $S = F_N^H D F_N$ para cierta matriz diagonal D . Vemos que la expresión de un elemento de S sería

$$S_{k,l} = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i k n / N} \lambda_{D,n} e^{-2\pi i l n / N} = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i (k-l) n / N} \lambda_{D,n}.$$

Si calculamos otro elemento de la misma diagonal (s filas y columnas más):

$$\begin{aligned} S_{(k+s) \bmod N, (l+s) \bmod N} &= \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i ((k+s) \bmod N) n / N} \lambda_{D,n} e^{-2\pi i ((l+s) \bmod N) n / N} = \\ &= \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i ((k+s) \bmod N - (l+s) \bmod N) n / N} \lambda_{D,n} = S_{k,l}; \end{aligned}$$

y por tanto la matriz es circulante. El razonamiento inverso es análogo. ■

De esta demostración podríamos entrever que existe una relación entre la primera columna de S , s , y su respuesta en frecuencias. En efecto, un simple cálculo nos mostrará el siguiente resultado.

Teo. 9.5- La respuesta en frecuencias de S viene dada por

$\lambda_S = \sqrt{N} F_N s$,
donde s es la primera columna de S . Y al contrario:

$$s = \frac{1}{\sqrt{N}} (F_N)^H \lambda_S.$$

Podemos reescribir la primera de estas expresiones como

$$\lambda_{S,n} = \sum_{k=0}^{N-1} s_k e^{-2\pi i n k / N}, n = 0, 1, \dots, N-1;$$

donde $s = (s_k)_{k \in [0, N-1]}$ es la primera columna de S . Si tenemos filtros en los que la mayoría de los coeficientes son 0, podemos encontrar expresiones muy simples. Por ejemplo, con el filtro que usamos previamente teníamos

$$s_0 = \frac{1}{2}, s_1 = \frac{1}{4}, s_{-1} = \frac{1}{4};$$

y por tanto

$$\begin{aligned} \lambda_{S,n} &= \frac{1}{2} e^0 + \frac{1}{4} e^{-2\pi i n / N} + \frac{1}{4} e^{-2\pi i n (N-1) / N} = \\ &= \frac{1}{2} + \frac{1}{4} e^{-2\pi i n / N} + \frac{1}{4} e^{2\pi i n / N} = \frac{1}{2} + \frac{1}{2} \cos\left(\frac{2\pi n}{N}\right). \end{aligned}$$

De esto podemos generalizar:

Teo. 9.6- La función $\lambda_S(\omega)$ definida en $[0, 2\pi)$ por

$$\lambda_S(\omega) = \sum_k t_k e^{-i k \omega}, k \in [0, N-1],$$

donde los t_k son los coeficientes del filtro, satisface

$$\lambda_{S,n} = \lambda_S\left(\frac{2\pi n}{N}\right), n = 0, 1, \dots, N-1; \forall N.$$

Demostración. Para cualquier N tenemos

$$\begin{aligned} \lambda_{S,n} &= \sum_{k=0}^{N-1} s_k e^{-2\pi i n k / N} = \sum_{0 \leq k < \frac{N}{2}} t_k e^{-2\pi i n k / N} + \sum_{\frac{N}{2} \leq k \leq N-1} t_{k-N} e^{-2\pi i n k / N} = \\ &= \sum_{0 \leq k < \frac{N}{2}} t_k e^{-2\pi i n k / N} + \sum_{-\frac{N}{2} \leq k \leq -1} t_k e^{-2\pi i n (k+N) / N} = \sum_{-\frac{N}{2} \leq k < \frac{N}{2}} t_k e^{-2\pi i n k / N} = \lambda_S\left(\frac{2\pi n}{N}\right) \blacksquare \end{aligned}$$

A ambas expresiones las llamamos *respuesta en frecuencias*; $\lambda_{S,n}$ será el *vector de respuestas en frecuencia* y la nueva $\lambda_S(\omega)$ la *respuesta en frecuencias continua*.

Obs. 9.7- λ_S es periódica con periodo 2π , podemos restringir ω a $[-\pi, \pi)$. Valores cercanos a los extremos representan frecuencias altas, mientras que cerca de 0 están las bajas.

La respuesta en frecuencias de un vector (continua) es la que determina cómo se comportará un filtro. Cuantos más coeficientes (muestras) tengamos, más se aproximará el resultado de filtrar al caso ideal de filtro perfecto. Más sobre esto a continuación.

Hay una serie de propiedades sobre respuestas en frecuencia que merece la pena ver:

Teo. 9.8- Para la respuesta en frecuencias continua se cumple

- i) Satisface $\lambda_S(-\omega) = \overline{\lambda_S(\omega)}$.
- ii) Si S es un filtro, S^T también. Además, si $\lambda_S(\omega)$ es la respuesta en frecuencias de S , la de S^T es $\overline{\lambda_S(\omega)}$.
- iii) S simétrica implica λ_S real. Si S es antisimétrica, puramente imaginaria.
- iv) Si S_1, S_2 son filtros digitales, $S_1 S_2$ también y $\lambda_{S_1 S_2}(\omega) = \lambda_{S_1}(\omega) \lambda_{S_2}(\omega)$.

Def. 9.9- La convolución de dos vectores x, y es el vector con cada entrada

$$(x * y)_n = \sum_k x_k y_{n-k}.$$

Así, aplicar un filtro S a un vector equivale a buscar la convolución de la primera columna del filtro con el vector de entrada (extendido periódicamente). En la práctica se trabaja sólo con los coeficientes, no es útil formar y almacenar la matriz completa en memoria (recordemos que el número de muestras suele ser muy elevado).

A continuación veremos una serie de filtros básicos que conviene conocer. Tres de los más importantes son los siguientes:

Def. 9.10- Un filtro S es

1. Paso bajo ideal si: $\lambda_S(\omega) = 1$ cuando $|\omega|$ es baja, $\lambda_S(\omega) = 0$ cuando se acerca a π . Esto es, aniquila frecuencias altas y mantiene las bajas; a partir de cierta frecuencia.
2. Paso alto ideal: opera al contrario, aniquilando frecuencias menores a cierto valor y manteniendo las altas.
3. Paso banda ideal si $\lambda_S(\omega) = 1$ cuando $|\omega| \in [a, b] \subset [0, \pi]$, 0 en el resto.

Otro filtro relevante es el de retardo. Consiste en tener todos los coeficientes nulos, excepto uno de ellos que es 1. La matriz circulante correspondiente, E_d , tiene por tanto 1s en una diagonal y 0 en el resto. Su primera columna sería:

$$e_d = (0, 0, \dots, 1, \dots, 0, 0);$$

donde el 1 se encuentra en la posición d de este vector.

Def. 9.11- La matriz circulante E_d tiene como primera columna al vector e_d , y

$$(E_d x)_n = \sum_{k=0}^{N-1} (E_d)_{n,k} x_k = \sum_{k=0}^{N-1} (E_d)_{(n-k) \bmod N, 0} x_k = x_{(n-d) \bmod N}.$$

Es decir, este filtro atrasa una señal (vector) por d muestras: nos permite retrasar señales. La respuesta en frecuencias es $\lambda_S(\omega) = e^{-id\omega}$, de magnitud 1. Evidentemente es útil en muchos ámbitos, aunque en la práctica sería más fácil simplemente recolocar las muestras.

Como ejemplo, con él podemos formar un filtro que añada eco, simplemente teniendo la diagonal principal de I_N y una segunda con una cierta constante $0 < c \leq 1$. La matriz sería:

$$S = I_{N \times N} + c * E_d.$$

Tras escuchar la señal normalmente, d muestras más tarde volveríamos a escuchar lo mismo pero ligeramente atenuando (por estar multiplicado por c).

Veamos el desarrollo matemático en profundidad de un filtro de paso bajo, para entender el proceso. Queremos un filtro ideal, que mantenga frecuencias por debajo de ω_c y elimine el resto. Es muy importante recordar que aún no estamos hablando de frecuencias en Hz , pues no estamos considerando la frecuencia de muestreo. En el bloque de aplicaciones veremos este otro caso.

Buscamos que la respuesta en frecuencias sea un vector con 1 en las entradas $0, \dots, L$ y $N - L, \dots, N - 1$; 0 en el resto. Esta L vendrá de $\omega_c = 2\pi L/N$. Podemos usar ahora la expresión del teorema 9.5,

$$s = \frac{1}{\sqrt{N}} (F_N)^H \lambda_s.$$

Calculamos, sustituyendo por simplicidad $\delta = e^{\pi i n/N}$:

$$\begin{aligned} s &= \frac{1}{N} \left(\sum_{k=0}^L \delta^{2k} + \sum_{k=N-L}^{N-1} \delta^{2k} \right) = \frac{1}{N} \left(\sum_{k=-L}^L \delta^{2k} \right) = \frac{1}{N} \delta^{-2L} \frac{1 - \delta^{2(2L+1)}}{1 - \delta^2} = \\ &= \frac{1}{N} \delta^{-2L} \left(\delta^{-1} \delta^{2L+1} \cdot \frac{\delta^{-(2L+1)} - \delta^{2L+1}}{\delta^{-1} - \delta} \right) = \frac{1}{N} \frac{\sin\left(\frac{\pi n(2L+1)}{N}\right)}{\sin\left(\frac{\pi n}{N}\right)}. \end{aligned}$$

Tenemos que los N coeficientes están equiespaciados a lo largo de esta función. Lamentablemente, esta expresión no tiene soporte finito: necesitaríamos infinitos coeficientes para tener el filtro ideal. En la próxima sección veremos en qué se traduce esta limitación al filtrar una señal.

Por otro lado, en general los coeficientes del filtro serán no nulos, y calcularlos todos es computacionalmente costoso. En algunos casos, como en aplicaciones en vivo, es importante filtrar rápidamente y no siempre disponemos de los coeficientes previamente. La solución a este problema es no calcular algunos de ellos, tan sólo los más importantes:

$$\left\{ \frac{1}{N} \frac{\sin\left(\frac{\pi n(2L+1)}{N}\right)}{\sin\left(\frac{\pi n}{N}\right)} \right\}_{k=-N_0}^{N_0}. \quad (9.8)$$

Cuanto mayor sea N_0 , mejor funcionará el filtro, pero más tendremos que calcular.

Con esto queda terminado el aspecto teórico de los filtros. En el resto del documento veremos otra forma de implementar este filtro, y otras aplicaciones de la teoría de Fourier.

No debemos cometer el error de pensar que los filtros digitales son la única forma de trabajar con la transformada. Éstos tratan cada frecuencia individualmente, y por tanto ocupan tan sólo una fracción del total. Aun así, el diseño de filtros es un tema muy profundo y que merece ser estudiado extensivamente.

Aplicaciones

A pesar de la complejidad matemática tras la transformada de Fourier, en el sentido práctico la idea es simple: dada una señal, podemos calcular su espectro para obtener nueva información. Esto tiene aplicaciones de todo tipo y en casi cualquier campo, haciéndola una herramienta indispensable y la base de incontables técnicas modernas.

Aquí veremos usos en la edición y el análisis de imagen y sonido. Éstos deberían darnos una idea de la variedad de aplicaciones en las que está presente esta teoría, si mantenemos en mente que son una parte muy pequeña del total.

Casi cualquier aplicación del análisis de Fourier consiste en tres pasos: transformar la señal original mediante la FFT o alguna de las variantes, modificar y adquirir la información deseada del espectro, y deshacer la transformación con el algoritmo correspondiente. Es gracias a este poder ir y venir al plano de frecuencias que tenemos tantos usos posibles, siendo los filtros el ejemplo más simple.

Ya vimos que no podemos aplicar los resultados disponibles a cualquier señal, necesitan cumplir ciertos requisitos. Por suerte, las señales físicas se comportan bien y podemos usar el análisis de Fourier; suelen tratarse de funciones localmente integrables o continuas. Además, muchas de ellas son finitas y pueden extenderse periódicamente, y casi siempre son de variación acotada.

Aquí trabajaremos con vectores de muestras y no señales continuas, pues no podemos por ejemplo almacenar todo el audio, sólo su valor cada cierto tiempo. Por tanto nos centraremos en las versiones discretas de la transformada, cuyos usos además son especialmente prolíficos. Aun así, debemos mantener en mente los resultados sobre la versión continua y el muestreo. En el apéndice *B* tenemos el código comentado, y en [27] podemos encontrar una carpeta *online* con ejemplos de lo obtenido.

10. Sonido

Desarrollar técnicas de análisis de señales requiere un conocimiento sobre el medio en el que trabajamos; necesitamos saber qué cualidades aprovechar y en qué se traducen nuestros cambios. Como dijimos, el sonido no es más que una variación en la presión del aire a través del tiempo, que podemos tratar como una señal unidimensional. La clave en este caso se encuentra en que la altura del sonido, que sea más grave o agudo, viene dada por la frecuencia de estas vibraciones.

Así, la transformada de Fourier de señales auditivas se traduce a descomposición en sonidos de distintas alturas. Es por ello que la acústica es el campo más evidente para usar análisis de Fourier. A partir de aquí se trata de idear y diseñar distintas formas de trabajar con esta información. Veremos con *Matlab* una serie de técnicas ordenadas por dificultad para trabajar con sonido, a medida que nos familiarizamos con este medio.

Hay muchas formas de almacenar audio, pero nosotros trabajaremos con *.wav*. Es el formato más indicado en nuestro caso: contiene el valor de la frecuencia de muestreo y el conjunto de muestras. De esta forma partiremos de un conjunto de valores en el tiempo y podremos comenzar a filtrar y transformar directamente. Además, no tiene pérdida de calidad, pero al mantener tanta información no se usa normalmente. En *Matlab* podemos abrir este formato para obtener toda la información, con las muestras en un vector (o dos, si el sonido fuera *Stereo*); en nuestro caso siempre será *mono*.

10.1. Filtro de paso bajo:

La primera aplicación que veamos será el filtro de paso bajo para señales de sonido. Recordemos que este filtro buscaba eliminar las frecuencias a partir de cierta cota; traduciéndolo a esta situación: sonidos de más de cierta altura. Es un filtro básico y que se utiliza asiduamente.

En la sección 9 vimos cómo desarrollar este filtro calculando sus coeficientes explícitamente. En concreto, en (9.8) teníamos la expresión general. Si quisiéramos, nuestro código podría ser tan simple como tomar estos coeficientes del filtro, formar la matriz correspondiente y multiplicarla por el vector de muestras del audio. Dado que esto tiene poco valor ilustrativo, siendo el cálculo de los coeficientes el aspecto más relevante, veamos otra forma de hacerlo.

En lugar de formar el filtro en sí y filtrar la señal, transformaremos paso a paso el vector de muestras. Esto debería ayudarnos a entender mejor qué es realmente un filtro digital. La idea es hacerlo en tres pasos: calcular la transformada de las muestras, modificar como queramos el resultado y finalmente deshacer la transformación.

Dado que poseemos un conjunto de muestras, utilizaremos la DFT para el primer paso, utilizando por ejemplo

$$y_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-2\pi i n k / N};$$

donde N es el número de muestras y x_k cada una de ellas. Se trata de la fórmula introducida en la sección 5, y que como dijimos nos dejará un conjunto de coeficientes correspondientes a cada uno de los tonos digitales puros. Por eficiencia, podemos usar alguna implementación de la FFT para obtener lo mismo pero rápidamente. Matlab tiene ya su propia FFT, pero el proceso sería idéntico utilizando las expresiones introducidas en el teorema 6.1. En cualquier caso, obtendremos un vector del mismo tamaño que el de muestras, y cada posición indicará la altura correspondiente a dicha frecuencia.

En la figura 3 mostrada a continuación podemos ver el espectro de un audio de un segundo de ruido blanco; una señal donde cada muestra es una variable independiente dada por una distribución uniforme definida en cierto intervalo. Esto resulta en una presencia muy similar de cada frecuencia. Las muestras fueron tomadas con una frecuencia de 44100Hz , y debido al teorema de Shannon no consideramos las frecuencias más allá de 22050Hz .

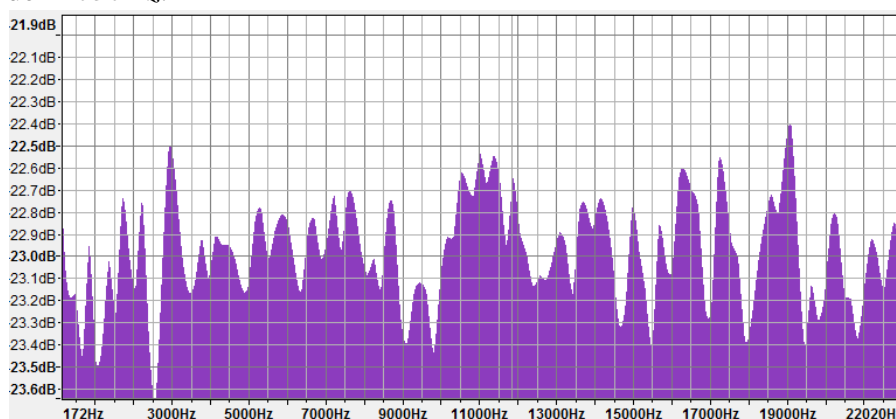


Figura 3: Espectro de ruido blanco.

Ahora que estamos en el plano de frecuencias, podemos modificar cada una de ella como queramos. Evidentemente, no se trata de un espectro continuo, sino un conjunto de valores, pero dada la gran cantidad de muestras no tendremos problemas en este sentido. Nótese que este espectro representa el módulo del coeficiente asociado a cada frecuencia, y no nos dice nada sobre el ángulo, pues aquí no nos interesa.

En este caso, buscamos eliminar las frecuencias a partir de cierto punto, por ejemplo la mitad superior (a partir de 11025Hz). Para ello simplemente cambiamos a 0 las entradas de la segunda mitad del vector de la transformada. Suprimimos así la presencia de esas frecuencias, manteniendo igual el resto.

El último paso es calcular la IDFT del vector modificado:

$$x_k = \sum_{n=0}^{N-1} y_n e^{2\pi i n k / N}.$$

O, si preferimos hacerlo de forma óptima, utilizamos la IFFT con el mismo fin. En cualquier caso, habremos obtenido el audio filtrado.

En la figura 4 vemos cómo quedaría el espectro de la señal filtrada. No hay frecuencias presentes de más de 11025Hz aproximadamente, como era de esperar; pero la caída de las frecuencias es gradual. En la sección 9 ya mencionamos que era imposible tener un filtro ideal, pues serían necesarios infinitos coeficientes. Esto hace que no podamos tener un cambio instantáneo.

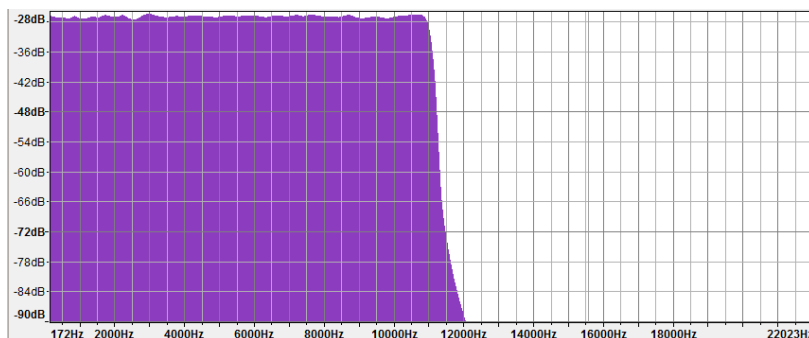


Figura 4: Espectro del ruido blanco tras ser filtrado.

Este fenómeno se produce comúnmente con filtros cuya respuesta en frecuencias no es continua y cambia repentinamente. Se dice que el filtro requiere una *región de transición*. Este fenómeno es muy relevante, y una pieza clave en el diseño de filtros. Por ello existen muchos filtros de paso bajo, como el de *Butterworth*, cuyas respuestas en frecuencia son más regulares y producen mejores resultados.

El programa *lowpass.m* realiza todo el proceso: dados un archivo de sonido (en *.wav*, grabado a 44100Hz) y una frecuencia, elimina sonidos de altura mayor a ésta. Como ejemplo podemos escuchar los archivos [pitido.wav](#) y [result Lowpass.wav](#)⁵, en los que eliminamos un pitido, aunque el audio resultante queda ligeramente trastocado (estamos eliminando parte de la información). En la práctica seríamos más precisos y filtraríamos sólo la banda de frecuencias correspondiente (en lugar de eliminar coeficientes a partir de cierto punto, descartamos un intervalo).

Nótese que el audio resultante tendrá un volumen aparentemente menor. Esto se debe a que estamos eliminando buena parte de la energía de la señal; y es fácilmente compensable reescalando la transformada (multiplicando los valores).

⁵ Son *links*, pueden tardar algo en cargar. El resultado se ha obtenido con nuestro código.

10.2. Compresión de Sonido

La compresión de datos es un campo donde la transformada de Fourier es vital, casi cualquier formato la utiliza en alguna de sus variantes. Aquí tan sólo hablaremos de compresión con pérdidas (*lossy* en inglés), pues la mayoría de técnicas de compresión sin pérdidas se basan en un mejor aprovechamiento de los datos. Para recalcar la importancia de estas técnicas; sin compresión de datos no podríamos enviar datos digitalmente fácilmente, o no tendríamos la telefonía móvil actual, entre muchas otras cosas.

Si no es necesario mantener exactamente la información, la idea es descartar parte para reducir el peso. En audio, no podemos desechar muestras para comprimir un archivo, pues nos quedaría algo diferente. Pero aquí entra en juego la transformada: podemos descartar frecuencias imperceptibles o poco relevantes, de forma que tenemos audio casi idéntico pero una reducción notable de peso.

Querremos aprovechar cuestiones físicas y psicológicas para minimizar la diferencia auditiva entre comprimir o no. Por ejemplo, podemos descartar frecuencias que no estén entre 20Hz y 20000Hz . Sabemos que el oído humano no es capaz de percibir las, y eliminarlas no causará ninguna diferencia para nosotros. Evidentemente, esto solo se aplica si tenemos frecuencias fuera de este intervalo, lo cual depende de la frecuencia de muestreo.

En cualquier caso, la fuente principal de ahorro de información es que tan sólo percibimos las frecuencias que se encuentren en mayor medida; esto es, si ciertas frecuencias tienen una amplitud notablemente menor al resto, pueden ser eliminadas sin provocar una pérdida relevante en calidad.

Buscamos descartar tantos coeficientes como podamos, manteniendo una calidad aceptable: nos quedaremos con los más importantes. Una primera forma de hacer esto sería tomando la transformada de las muestras normalmente y manteniendo sólo las de mayor módulo. Si lo hiciéramos así, veríamos que es complicado descartar coeficientes sin perder calidad. Esto se debe a que cada frecuencia tiene un efecto durante todo el audio, y eliminarlas deteriora notablemente el sonido. A lo largo de la señal, comenzarán y cesarán distintos sonidos de distintas alturas, y por ello muchas frecuencias tienen importancia, y no representan sonidos individuales. Aunque Fourier nos dé información sobre la señal, no es la que nos conviene.

Para combatir esto, usaremos una técnica básica: dividir la señal en bloques de corta duración. Así, sí podremos entender que las frecuencias representan los sonidos presentes en ese intervalo de tiempo, y por tanto descartar más coeficientes sin temor a perder el significado del audio original. No queda inmediatamente claro cómo funciona este método, pero lo veremos paso a paso a continuación.

Por último, también podemos utilizar una menor cantidad de *bits* para almacenar cada valor, y aunque así reduciríamos el número de tonos representables, si lo hacemos de forma razonable no notaremos la diferencia. En los *scripts compress.m* y *decompress.m* tenemos una posible implementación de todo esto. El primero de estos archivos transformaría un conjunto de muestras (audio en formato *.wav*) a uno codificado (*.mat*) que ocupa mucho menos espacio. El segundo sería el encargado de tomar el fichero codificado y reconstruir el audio para escucharlo.

Veamos el proceso de compresión.

i) Comenzamos dividiendo el vector de muestras en bloques de tamaño n . Por defecto, en nuestro código los tomamos de 2048 muestras, potencia de 2 para aprovechar el funcionamiento de la FFT. Nótese que por tanto cada bloque dura unas centésimas de segundo, considerando una frecuencia de muestreo de 44100Hz.

ii) Buscaremos ahora la transformada de cada bloque. Podríamos hacerlo con la DFT o la FFT normalmente, con las expresiones indicadas en las secciones correspondientes, pero es una perfecta ocasión para introducir un nuevo algoritmo: la **DCT**. Veamos en qué consiste.

La DCT, o *transformada de coseno discreta*, es una variante de la DFT para muestras reales: expresando la función como suma de cosenos, en lugar de senos y cosenos. Partiendo de un vector de reales, tenemos varias formas de extenderlo periódicamente. Si formamos una extensión simétrica respecto al 0, al calcular su transformada nos quedará que los coeficientes de los senos son 0 por ser antisimétricos: sólo quedarán cosenos.

Así, con la mitad de coeficientes obtenemos la misma precisión. Además, dado que las muestras son reales y la extensión es simétrica, tenemos que una mitad de los coeficientes es igual a la otra, y por tanto esta DCT necesita un cuarto de los coeficientes de la DFT usual. Evidentemente, se trata de un algoritmo idóneo para la compresión de datos.

Cabe mencionar que existen varios tipos de DCT (un total de 8) en función de la extensión de los datos⁶. Cada una tiene propiedades distintas, y según la situación se usa una u otra. En nuestro caso, y la más popular, utilizaremos la DCT-II; cuya expresión para un vector de muestras x_n es:

$$y_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right), k = 0, \dots, N-1;$$

donde y_k será el vector de la transformada resultante. No veremos el desarrollo formal hasta esta expresión, pues es análogo al de la DFT. Con esto calcularemos los coeficientes de la transformada de cada bloque, 2048 de ellos, y podemos empezar a modificarlos.

iii) Buscamos descartar tantos coeficientes de estos como podamos, quedándonos con los más importantes. En nuestro código, mantenemos 500 de los valores y los codificamos con 14 bits. Estamos desechando más de un cuarto de la información y reduciendo su peso; aunque pueda parecer excesivo veremos que el resultado obtenido es muy similar al original. Siempre podemos mantener más coeficientes para perder menos información, y de hecho diseñar correctamente un algoritmo de compresión depende en buena parte de esta decisión.

⁶ Por cada extremo del conjunto de muestras, tenemos dos decisiones: si tomar extensión par o impar, y si queremos hacer los datos (anti)simétricos respecto a una de las muestras o respecto al punto medio entre la muestra y su extensión. Por ejemplo, para la secuencia 1234 podemos extenderlo hacia la derecha con 1234321, o con 12344321. Esto nos deja 16 posibles extensiones, de las cuales las 8 simétricas son de la DCT y las antisimétricas de la DST (*transformada en senos discreta*).

En general nos conviene usar la DCT, por dar más libertad en los extremos (no es necesario que el vector de muestras se anule en ellos).

iv) Volviendo a nuestro algoritmo, tan sólo nos queda guardar los coeficientes deseados junto a sus índices (un vector por bloque). Por cada coeficiente a mantener necesitaremos un valor codificado en el número de *bits* elegido y otro que indique su posición en el vector. Por otra parte, también necesitamos guardar tres valores para descomprimir: la frecuencia de muestreo, el número de *bits* usados para codificar y el número de bloques que tenemos.

Para descomprimir, el proceso es intuitivo. Veamos cada paso:

i) Tras leer la información del archivo comprimido, queremos reconstruir la transformada con los coeficientes que guardamos: un vector por cada bloque que teníamos. En nuestro caso, partiendo de vectores de 2048 posiciones colocaremos los 500 coeficientes donde nos indiquen los índices, dejando nulo el resto.

ii) Ya tenemos la transformada de cada ventana, queremos ahora reconstruir el audio. Para ello calculamos la inversa, en nuestro caso con la IDCT:

$$x_n = \sum_{k=0}^{N-1} y_k \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right), n = 0, 1, 2, \dots, N-1;$$

donde N es el número de coeficientes (para nosotros 2048), y_k es el vector de coeficientes de la transformada y x_n el de muestras que buscamos.

iii) Sólo nos falta concatenar las muestras de cada bloque para obtener el audio completo. La frecuencia de muestreo a usar para reproducirlo será la misma que teníamos originalmente, y que también guardamos.

Es importante mencionar que este paso producirá ciertos defectos en el audio. Al pasar de un bloque al siguiente, tendremos tonos no nulos distintos entre ambos. Así, se producirán cambios leves pero repentinos. Aun así, en general este fenómeno no es perceptible, pero conviene saber que está presente.

Hemos utilizado con éxito una variante de la transformada de Fourier para reducir considerablemente el peso de un archivo de audio. Como ejemplo, tenemos [*classic.wav*](#) y [*compr.mat*](#). Pasamos de 18MB a 3, y si escuchamos el original y la versión decodificada ([*decomp.wav*](#)) veremos que no hay diferencia perceptible. En los formatos más comunes (*mp3*, por ejemplo) se utilizan técnicas muy similares a las aquí mostradas, aunque por supuesto optimizadas para maximizar la compresión.

El dividir en bloques la señal para su posterior análisis con Fourier es una de las herramientas más utilizadas cuando se busca más precisión. La próxima aplicación lo utilizará en más profundidad, y comprenderemos su potencial.

10.3. Cambiar duración del sonido

La última aplicación que veremos para sonido, y la más complicada, será cómo cambiar la duración de un fragmento de audio. La primera forma de abordar el problema sería modificando la frecuencia de muestreo con la que leemos el archivo. Así, podríamos por ejemplo reproducir el doble de muestras por unidad de tiempo para reducir la duración a la mitad.

Pero pronto nos daríamos cuenta que de este modo cambiamos también la altura del sonido, un efecto secundario indeseado en general. ¿Existe alguna forma de, a partir de las muestras, cambiar la duración sin afectar el tono? La respuesta es afirmativa, utilizando un algoritmo basado en el análisis de Fourier.

La idea es relativamente simple: buscaremos un nuevo conjunto de muestras a partir del original, de forma que mantengamos la altura y el audio, pero lo dilatemos (o contraigamos) en el tiempo. Para ello utilizaremos la *STFT*, transformada de Fourier de tiempo corto; veamos en qué consiste.

Cuando uno estudia señales con el análisis de Fourier, eventualmente se percata de que la transformada usual está limitada en ciertos aspectos. Dada una señal, su transformada nos indica su espectro, y poseemos un resultado análogo discreto. Pero aunque no perdamos información al transformarla, no podemos por ejemplo apreciar características temporales: no sabremos cuándo se produce algún cambio o cuando empieza o acaba el audio. Asimismo, no podemos utilizar esto para señales que estemos recibiendo en directo, pues necesitamos acceder a la función en su totalidad.

Así, nos gustaría tener una herramienta capaz de afrontar estos problemas. Una solución es la utilizada previamente: en lugar de tomar la transformada de toda la señal, analizar bloques de tamaño fijo a través del tiempo; estudiaremos la señal tanto temporalmente como en el espacio de frecuencias.

Mientras que antes nos bastó con tomar la DCT de cada bloque, ahora necesitamos realizar otras operaciones, y es necesario un estudio previo. Matemáticamente, tomar un bloque concreto no es más que multiplicar la señal $f(x)$ por una función llamada *ventana*. La natural es la función característica de ese intervalo ($\chi_{[-A,A]}$ por ejemplo). Pero, por la proposición 7.16, se tiene

$$\hat{g}(x) = (f \cdot \widehat{\chi_{[-A,A]}})(x) = \hat{f}(x) * \left(\frac{\sin(2\pi Ax)}{\pi x} \right).$$

Así, truncar la señal se traduce en convolucionar el espectro original con un seno cardinal. Cuanto menor es A , mejor es la aproximación de \hat{f} por \hat{g} , pero los cálculos son cada vez más costosos. Esto se debe al lento decrecimiento del seno cardinal, y su forma no se comporta bien con la convolución.

Esto nos lleva a sustituir la función característica por otras ventanas más convenientes; lo cual debería recordarnos a los problemas del filtro de paso bajo ideal. No entraremos aquí a los distintos tipos que existen, pues son muchos y con distintas características, apropiadas para cada situación. En [6] podemos encontrar otras opciones. Nosotros utilizaremos una de las más comunes, la ventana de *Hanning*:

$$\omega(t) = \frac{1}{2} \cdot \left(1 - \cos\left(\frac{2\pi t}{A}\right) \right);$$

que en ocasiones se expresa multiplicada por $\chi_{[-A,A]}$.

Desplazando esta ventana por la función y calculando la transformada obtenemos información local sobre el espectro de la función. Así, tendremos una familia de coeficientes que dependen de dos variables:

$$W_f(\lambda, b) = \int_{-\infty}^{+\infty} f(t) \bar{w}(t - b) e^{-2\pi i \lambda t} dt$$

Esta transformación, de f a $W_f(\lambda, b)$ es la llamada STFT. En esencia, nos indica cómo se comporta la función en el instante b de tiempo para la frecuencia λ . Perder precisión es inevitable, pero nos compensa el poder estudiar así las funciones. Para un estudio en más profundidad de la STFT y de su uso para el análisis de audio véanse [12] y [21].

Ahora que conocemos el aspecto teórico, volvamos a nuestro problema (discreto) y veamos cómo se utiliza la STFT en estos casos. Si lo que tenemos es un conjunto de muestras, comenzamos agrupándolas en bloque que multiplicamos por la función ventana (su valor en ciertos puntos). Cuanto mayor sea el grupo, mejor resolución tendremos en el plano de frecuencias, pero peor temporalmente, y viceversa.

Estos bloques se solapan en cierto número de muestras para evitar problemas, y para estudiar mejor la evolución a través del tiempo. Por último calculamos la FFT de cada uno de estos vectores, y guardamos los resultantes. En resumen:

$$W_x(\lambda, m) = FFT \left(\sum_{n=0}^{N-1} x(mh + n) w(n) \right) (\lambda);$$

donde x es el vector de muestras, m el bloque que estamos calculando, N es el ancho de los bloques, h el número de muestras entre bloques y $w(n)$ son las muestras de la función ventana.

Aún no es inmediatamente claro cómo usar esta nueva transformación para modificar la duración de un audio, lo veremos a continuación. Como ejemplo concreto, multiplicaremos la duración por $4/3$, y otros casos son análogos. Veamos paso por paso cómo; en el apéndice encontramos la implementación en *dilate.m*, *stft.m*, *interp.m* y *istft.m* con comentarios para mejor comprensión.

i) Comenzamos calculando la STFT de las muestras. Tomaremos bloques de 1024 muestras, considerando audio grabado a unos 20000Hz de frecuencia de muestreo; potencia de 2 para aprovechar la FFT. Como función de ventana utilizaremos Hanning, con 256 muestras entre cada bloque (un 75% de solapamiento)⁷.

A continuación, multiplicamos elemento a elemento el vector de muestras de cada bloque por los valores de la ventana; si v_b es el vector del bloque y N su tamaño:

$$\sum_{n=0}^{1023} v_b(n) w(n).$$

⁷ La anchura del bloque es un valor muy relevante, y para cuya elección se deben estudiar distintas opciones. Cuantas más muestras ocupe mejor resolución tendremos en la transformada (más coeficientes disponibles por bloque), pero peor resolución tendremos en el tiempo: se hará más difícil distinguir sonidos. Este ancho depende en buena parte de la frecuencia de muestreo usada para grabar el audio. Asimismo, cada tipo de ventana (y el número de muestras solapadas) afectan de formas distintas en cada caso, y estudios más profundos son precisos para una implementación óptima.

Nos quedarán vectores de 1024 muestras correspondientes al producto de la señal por la ventana. Ahora tomamos la FFT de cada uno de estos vectores para obtener la transformada de cada bloque modificado (por la ventana).

ii) En este punto, con la matriz obtenida de la STFT, será cuando modifiquemos la duración. Idealmente, tomaríamos cada sonido individualmente y cambiaríamos su duración. Eso evidentemente no es posible, pero con la STFT tenemos información local del espectro del audio, y podemos intentar extenderla. Partiendo de un total de N bloques, interpolaremos un nuevo conjunto de transformadas: $c \cdot N$ de ellas, si queremos multiplicar la duración por c . Podemos suponer qué ocurre entre bloques y construir una nueva matriz.

Un detalle de este método es que podemos aproximar fácilmente los módulos de los coeficientes, pero los ángulos (la fase de las funciones trigonométricas) son más complicadas. Por suerte, el resultado es menos sensible a cambios en la fase, aunque preferiríamos minimizar ruidos. Veamos cada cosa por separado.

Para los módulos, partiendo de un total de N vectores de complejos $u_m, m \in [1, N]$; queremos obtener $4/3 \cdot N$ vectores de reales $v_k, k \in [1, 4N/3]$. Sean

$$n_k = 3k/4; n_f = \lfloor n_k \rfloor; d = n_k - n_f;$$

queriendo calcular v_k . Entonces

$$v_k = (1 - d) \cdot |u_{n_f}| + d \cdot |u_{n_f+1}|.$$

Así tendremos una aproximación de los módulos entre dos de los bloques originales.

Para la fase, intentaremos aproximar cuál es la que corresponde a cada nuevo bloque, por orden. La primera será la misma que la de u_0 , pues parten del mismo punto. De aquí en adelante, guardaremos cuál es la fase hasta hora y añadiremos la diferencia entre los ángulos de u_{n_f} y u_{n_f+1} tras cada paso. Esto mantendrá la fase relativamente cercana a la que habría en una dilatación o contracción perfecta, y nos basta en nuestro caso.

Una fase incorrecta provoca ruido en forma de saltos entre los bloques tras reconstruir, y por ello son necesarios buenos algoritmos para tener sonido fluido. Con esto ya tendríamos módulo y ángulo de los complejos de la transformada modificada.

iii) Para terminar sólo nos falta invertir la STFT. El procedimiento es idéntico pero al contrario: empezamos calculando la IFFT de cada bloque de la transformada, obteniendo así un nuevo conjunto de vectores. Para cada uno de estos vectores, lo multiplicaremos por una nueva función ventana (llamada *ventana de síntesis*); en nuestro caso utilizaremos Hanning de nuevo. Finalmente, iremos sumando el resultado en un vector que contendrá el audio final que inicializamos con ceros. Empezando en la primera posición, sumaremos en las 1024 posiciones siguientes, avanzando 256 posiciones tras cada suma.

Con esto habremos terminado: tendremos un vector de muestras que contendrán el audio original dilatado en el tiempo. Como ejemplo, podemos escuchar [voice.wav](#) y [voice_longer.wav](#), obtenidas con nuestro código. Nótese que se produce algo más de ruido, aunque la diferencia es aceptable. Con algoritmos más complejos podemos modificar la duración sin un cambio perceptible.

Usando esta técnica podemos también cambiar fácilmente la altura del sonido, sin modificar la duración. Para ello nos basta con primero cambiar la duración acordemente y después reproducirlo a una frecuencia de muestreo mayor o menor. Como ejemplo, podemos escuchar [voice_higher.wav](#).

11. Edición Imagen

Pasemos ahora al estudio de imágenes, y el uso de Fourier para editarlas. Debemos comenzar familiarizándonos con el nuevo entorno, siendo el cambio evidente que ahora tenemos dos dimensiones. A efectos prácticos, los resultados pueden ser extendidos sin problema, pero con funciones trigonométricas bidimensionales. No entraremos aquí al desarrollo de los resultados, y nos bastará con las expresiones que veremos a continuación.

Digitalmente, una imagen puede ser representada por un conjunto de valores que indican el color de cada píxel. Así, en color tenemos tres valores por píxel (rojo, azul y verde) y podemos expresarlo con tres matrices, cuyos valores se encuentran entre la intensidad mínima y máxima (0 y 255, por ejemplo). Nosotros trabajaremos con imágenes en blanco y negro por simplicidad, y de este modo sólo tendremos que considerar una matriz, que indicará la intensidad de gris en cada punto.

Estos valores de intensidad pueden entenderse como muestras de la señal original: el objeto de la imagen. Así, el procedimiento es similar al de sonido, pero en este caso en dos dimensiones. Introducimos pues la DFT bidimensional:

$$F(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-2\pi i(um/M + vn/N)}.$$

Vemos que la expresión es muy similar a la usual, pero en este caso el espacio de entrada y de salida es \mathbb{C}^2 . De hecho, en esencia consiste en tomar varias transformadas seguidas: considerando la matriz de muestras, su FFT se puede obtener calculando la transformada de cada columna y después de cada fila del resultado.

La otra diferencia importante es que en este caso la imagen es una función del espacio y no del tiempo. Esto nos deja con unas propiedades y opciones completamente diferentes, y nos permite apreciar mejor las diferencias.

Antes de comenzar con los filtros, veamos un ejemplo de la apariencia de la transformada de una imagen:



Figura 5: Una imagen en blanco y negro y su transformada desplazada.

La de la izquierda es la original, *Lenna*, la imagen de prueba más comúnmente utilizada para este tipo de operaciones. A la derecha tenemos su transformada desplazada, en el centro los coeficientes correspondientes a frecuencias bajas y hacia el borde más altas. Para obtener esta transformada necesitamos trasladar los coeficientes para tener $F(0,0)$ en el centro, y la intensidad de cada píxel corresponde al módulo de los coeficientes $F(u, v)$.

Nótese como los valores en torno al centro son mayores; esto nos indica que las bajas frecuencias contienen más información que las altas, y de hecho ocurre en casi cualquier imagen. Las rectas correspondientes a $F(0,v)$ y $F(u,0)$ son también más importantes, y contienen más información que el resto.

11.1. Filtro de paso bajo

El filtro de paso bajo opera igual en imágenes que en sonido, pero en este caso puede ser más interesante por tener un efecto más notable. Queremos mantener coeficientes correspondientes a frecuencias menores a cierto valor, descartando el resto. Veamos el proceso, partiendo de una imagen cualquiera (una matriz de intensidades).

En primer lugar necesitamos tomar la transformada discreta de esta matriz. Para ello, calculamos la FFT usual en vectores: primero la transformada de cada columna, quedándonos una nueva matriz, y después de cada fila. Así, obtendremos la matriz de coeficientes correspondiente a la imagen.

Como decíamos antes, es conveniente trasladar los elementos de la matriz para tener $F(0,0)$ en el centro. Para ello sólo necesitamos intercambiar el cuadrante superior izquierdo de la matriz con el inferior derecho y el superior derecho con el inferior izquierdo.

Podemos ahora eliminar las frecuencias indeseadas. Para ello, formamos una matriz con unos en un círculo en torno al centro, y nula en el resto. Si multiplicamos cada elemento de esta matriz por los de la transformada, nos quedará lo que vemos en la figura 6: tan sólo nos quedan las frecuencias centrales.

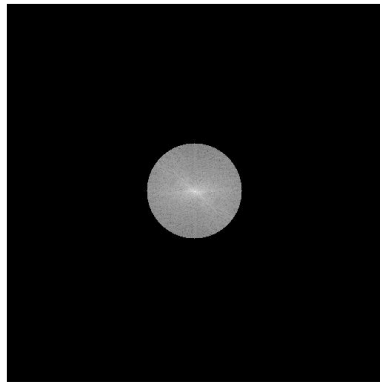


Figura 6: Transformada tras eliminar frecuencias altas.



Figura 7: Transformada tras eliminar frecuencias altas.

Finalmente, nos queda calcular la transformada inversa. Debemos primero deshacer la traslación usada para centrar $F(0,0)$. Tras esto, usamos por ejemplo

$$f(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{2\pi i(mu/M + nv/N)};$$

la IDFT en dos dimensiones, para encontrar la imagen filtrada. De nuevo podemos entenderlo como la aplicación sucesiva de varias IDFTs unidimensionales. En la figura 7 encontramos el resultado, del cual podemos realizar varias observaciones.

Lo más llamativo es cómo a pesar de haber suprimido la gran mayoría de los coeficientes, la imagen es muy similar. Esto se debe a que, como ya observamos, son las frecuencias bajas las que contienen la mayor parte de la información en las imágenes.

Como era de esperar, eliminar altas frecuencias deja los contornos más suaves, ya que en ellos es donde se produce un cambio brusco que no puede representarse con frecuencias bajas. En cambio, las secciones de imagen con cambios graduales sí pueden serlo.

Análogamente, es interesante ver qué ocurre tras eliminamos las frecuencias bajas. Para ello, nos basta con multiplicar elemento a elemento la matriz de la transformada original por una contraria a la anterior: nula en un círculo central e igual a 1 en el resto. En la figura 8 vemos el resultado de este producto, y en la 9 la imagen modificada.

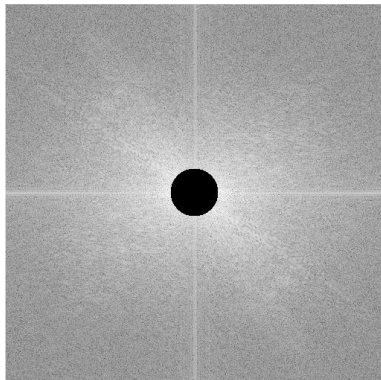


Figura 8: Espectro tras eliminar las frecuencias bajas.



Figura 9: Imagen filtrada, sin frecuencias bajas.

Lo obtenido refuerza lo ya dicho: eliminar las frecuencias bajas, incluso unas pocas como aquí, hace que se pierda la mayoría de la información. Nos quedan sólo los contornos, pues como dijimos son en esencia frecuencias altas.

Mientras que con sonido teníamos que cualquier frecuencia podía estar presente, pues representaban la altura, en imagen tenemos que algunas son más relevantes que otras y contienen mucha más información.

En el archivo *lospassIM.m* podemos ver una posible implementación de todo esto, tanto para filtros de paso bajo como altos. Matlab tiene implementaciones propias de la FFT bidimensional, y nos queda un código simple. Cabe recalcar que aunque pueda parecerlo por las figuras, no son filtros ideales; como ya dijimos esto es imposible.

Estos filtros son usados con fines tanto estéticos como técnicos. Es posible ver qué representa cada frecuencia en una imagen, y podemos por ejemplo eliminar algunos defectos como sombras o reflejos buscando a qué frecuencias corresponden. Además, como ocurría con sonido, es posible dividir las imágenes en bloques cuadrados y tratar cada uno por separado para una mayor precisión. En [8] (y [9]) tenemos un estudio exhaustivo de todo tipo de técnicas de edición de imagen basadas en Fourier.

11.2. Patrones en Imágenes

El reconocimiento de patrones es un campo relativamente moderno y cada vez más relevante. En esencia, consiste en encontrar secuencias y patrones en señales. Para ello se estudian características de la señal en cuestión, y se cuantifica la similitud con el patrón. Si este valor alcanza cierta cota se supondrá una respuesta afirmativa.

Así, en este campo es extremadamente importante una correcta elección de características para distinguir entre los distintos casos. Lo más importante al tomar esta decisión es en qué medida podemos reconocer cada situación estudiando esa característica.

La transformada de Fourier, en cualquier medio, nos permite pasar al plano de las frecuencias. Así, nos da una nueva dimensión de cualidades a las que acceder para diferenciar entre casos. Nos interesa que estas características sean lo más dispares posibles entre cada caso, y en muchas situaciones trabajar con frecuencias en lugar de muestras nos permite hacer esta clasificación con más precisión, o incluso es la única forma de hacerlo correctamente.

Lamentablemente, no sería posible desarrollar mucho estas técnicas aquí, pues son bastante complejas y requeriría tiempo explicarlas. En [15] podemos encontrar un ejemplo de clasificación usando análisis de Fourier. Dicho esto, sí podemos presentar un ejemplo para ilustrar estas ideas. El programa *recog.m* calcula la llamada *correlación* de dos imágenes. Esta operación es una forma de medir el grado de similitud entre dos señales, en nuestro caso imágenes.

En esencia, la correlación compara las frecuencias del patrón con cada sección del mismo tamaño. Cuanto mayor sea el grado de similitud entre patrón y sección, mayor será el valor resultante. Ocurre que la correlación en imágenes es equivalente a la convolución girando 180° la segunda imagen. Veamos paso por paso el proceso:

i) Debemos girar el patrón a localizar (una imagen) 180° , para prepararla para la correlación.

ii) Calculamos la FFT del patrón girado. Para ello procedemos como con el filtro de paso bajo, tomando varias FFTs unidimensionales consecutivas. Por último, es necesario añadir filas y columnas de ceros a la matriz (a la derecha y debajo) hasta que tenga el mismo tamaño que la imagen completa; veremos luego por qué. Llamemos P a la matriz resultante.

iii) Calculamos normalmente la FFT de la imagen original. Sea Q la matriz resultante.

iv) Dijimos que la correlación era equivalente a la convolución con la segunda imagen rotada 180° . Además, por la Prop. 7.16 tenemos que

$$\widehat{f \cdot g} = \hat{f} * \hat{g}.$$

Por tanto, nos basta con calcular el producto elemento a elemento de P y Q para obtener la transformada de la correlación de la imagen y el patrón. Por esto era necesario añadir ceros a la FFT del patrón, para poder multiplicarla así con Q .

v) Si calculamos la IFFT de la matriz resultante tendremos la correlación. Nos basta con varias IFFTs unidimensionales para ello, como dijimos. Sea C el resultado.

vi) Queremos localizar en qué puntos de la imagen original está presente el patrón. Para ello buscamos el máximo de C , que será donde ambas sean más parecidas. Buscamos todos los puntos con ese valor o ligeramente inferior, y será allí donde esté el patrón.

Es conveniente ver un ejemplo del resultado. Consideremos que queremos buscar instancias de la letra 'a' en una pieza de texto, ambas cosas como imágenes. Es un ejemplo muy simple pero que nos servirá para entender lo que queda. Tras ejecutar nuestro programa obtendríamos:

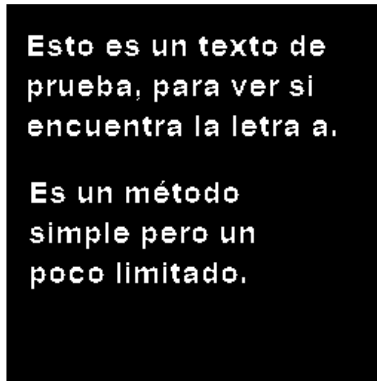


Figura 10: La imagen completa.



Figura 11: El patrón a buscar

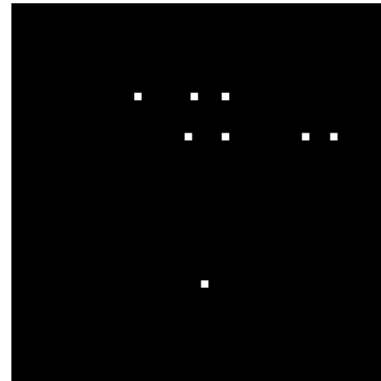


Figura 12: Los puntos donde la correlación es máxima.

La figura 10 es una imagen con un cierto texto, y la 11 una imagen de la *a*, que queremos encontrar. En la 12 vemos el resultado: como queríamos, cada aparición del patrón aparece marcada por un punto, correspondiente a máximos de la correlación.

Además, no hay falsos positivos, que sólo se darían en este caso si no hay ninguna aparición del patrón (para evitar este caso tendríamos que hacer estudios previos para determinar cuándo no está presente). Evidentemente, usar esta técnica para encontrar letras en un texto es muy ineficiente, pero para detectar fragmentos de otro tipo la correlación se utiliza frecuentemente.

11.3.Compresión de Imágenes

Para terminar, mencionemos los principios de la compresión de imágenes. Como en sonido, para ahorrar espacio las técnicas usadas se basan en buena parte en la transformada de Fourier. Un ejemplo relativamente simple de ello es el formato *JPEG*, que a pesar de su sencillez es uno de los más utilizados. Nótese como el proceso de compresión es muy similar al que utilizamos en audio.

El primer paso es dividir la imagen en cuadrados de 8 píxeles de lado. Cada uno de ellos se tratará independientemente, comprimiéndolos individualmente. Existen múltiples algoritmos para estas compresiones, pero en *.jpg* se usa de nuevo la DCT; aunque en este caso se utiliza la bidimensional. Nos basta con calcular la DCT de cada columna y la de cada fila resultante para obtenerla.

Dado que tenemos 64 coeficientes por bloque, al transformarlo nos quedan otros 64. Ahora sólo nos queda proceder como hicimos con el audio: reduciendo el número de bits para representar los coeficientes y descartando algunos de ellos. Guardamos los coeficientes escogidos con distintas técnicas informáticas para ahorrar aún más espacio y habremos comprimido la imagen.

Bibliografía

- [1] J.Bell. *The Poisson Summation Formula, the sampling theorem, and Dirac combs*. Department of Mathematics, University of Toronto (2014).
- [2] P. Bourke. *Fourier Method of Designing Digital Filters*. Paul Bourke (1999).
- [3] Luke Dodd. *Simple audio compression*. Disponible en: <http://www.lukedodd.com/really-simple-audio-compression/>
- [4] Dan Ellis. *A phase vocoder in Matlab*. Disponible en: <http://labrosa.ee.columbia.edu/matlab/pvoc/>
- [5] C.L. Epstein. *Introduction to the Mathematics of Medical Imaging*, 2ªed. Siam (2007).
- [6] C. Gasquet, P.Witomski. *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*, 1ª ed. Springer (1999).
- [7] E.F. Glynn. *Fourier Analysis and Image Processing*. Stowers Institute for Medical Research (2007).
- [8] R.C. Gonzalez, R.E. Woods. *Digital Image Processing*, 2ªed. Prentice Hall (1977).
- [9] R.C. Gonzalez, R.E. Woods. *Digital Image Processing using Matlab*, 1ªed. Prentice Hall (1977).
- [10] M.T. Heideman, H. Johnson, C.S. Burrus. *Gauss and the history of the Fast Fourier Transform*. IEEE ASP magazine (1984).
- [11] J. Guckert. *The Use of FFT and MDCT in MP3 Audio Compression*. University of Utah (2012).
- [12] R. Gutiérrez Osuna. *Introduction to Speech Processing*. Texas A&M University.
- [13] B. Kessler. *A "Sound" Approach To Fourier Transforms: Using Music to Teach Trigonometry*. Western Kentucky University (2007).
- [14] S.R. Kulkarni. *Frequency Domain and Fourier Transforms*. Princeton University (2002).
- [15] J. Lerma Aragón, J. Álvarez Borrego. *Vectorial Signatures for Pattern Recognition*. Intech (2012). Disponible en: <http://cdn.intechopen.com/pdfs-wm/35226.pdf>
- [16] G. Lerman. *The Shannon Theorem and Its Implications*. University of Minnesota (2015).
- [17] K. Mørken, Ø. Ryan. *Fourier analysis and applications to sound processing*. Universitetet i Oslo (2012).
- [18] K. Mørken, Ø. Ryan. *Wavelets and applications to image processing*. Universitetet i Oslo (2012).
- [19] L. Petracovici, J.M. Rosenblatt. *Introduction to Sampling*. University of Illinois (1998).
- [20] R.S. Stanković, J.T. Astola¹, M.G. Karpovsky. *Remarks on History of Abstract Harmonic Analysis*. Dept. of Computer Science, Faculty of Electronics, 18 000 Niš, Serbia.
- [21] J.O. Smith. *Spectral Audio Signal Processing*. Center for Computer Research in Music and Acoustics (2011).
- [22] S.W. Smith. *The Scientists and Engineer's Guide to Digital Signal Processing*. California Technical Publishing (2011).

- [23]J. Sueur. *A very short introduction to sound analysis for those who like elephant trumpet calls or other wildlife sound*. Museum national d'Histoire naturelle (2016).
- [24]M. Wakin. *Standard Test Images*. Disponible en: ece.rice.edu/~wakin/images/
- [25]A. Zisserman. *2D Fourier Transforms and Applications*. University of Oxford (2014).
- [26]EECS Dept. *The Nyquist-Shannon Sampling Theorem*. UC Berkeley (2016).
Disponible en: ptolemy.eecs.berkeley.edu/eecs20/week13/nyquistShannon.html
- [27]<https://drive.google.com/open?id=0B99rJ39IrExQOXVWVDE4Wm5tSzQ>

A. La Integral de Lebesgue

En este primer anexo introduciremos una serie de conceptos y resultados sobre la integral de Lebesgue que son necesarios para llegar a la transformada de Fourier. No buscamos realizar un desarrollo completo de este tema, pues es bastante profundo, y nos ceñiremos a los resultados necesarios, partiendo de teoría de la medida.

La motivación para introducir una nueva teoría de integrales es que la de Riemann está limitada en ciertos aspectos. Operar con límites o derivadas con la nueva integral será mucho más cómodo, gracias a algunos de los resultados que veremos.

Comencemos viendo cómo se define la integral de Lebesgue:

Def. A.1- Sea un espacio (X, Σ, μ) de medida, y una función $f: X \rightarrow \overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{+\infty\}$ medible y no negativa. La integral de f respecto a la medida μ viene dada por

$$\int_X f d\mu = \sup \left\{ \int_X e d\mu \mid 0 \leq e \leq f, e \text{ simple} \right\}.$$

Decimos que f es integrable si $\int_X f d\mu$ es finita.

Esta definición se basa en la integral de una función simple; dada $e = \sum_{i=1}^n \alpha_i \chi_{S_i}$, su integral es:

$$\int_X e d\mu = \sum_{i=1}^n \alpha_i \mu(S_i).$$

De la definición vemos claro que las propiedades básicas de la integral de Riemann como la linealidad y la monotonía se mantienen para funciones medibles no negativas. Si E es un conjunto medible, definimos la integral de f en E con

$$\int_E f d\mu = \int_X f \chi_E d\mu.$$

Para generalizar a cualquier función medible, nos basta con considerar las partes positiva y negativa: f_+, f_- .

Def. A.2- Sea un espacio (X, Σ, μ) de medida, y una función $f: X \rightarrow \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ medible. Se dice que f es integrable en E si las integrales $\int_E f_+ d\mu$ y $\int_E f_- d\mu$ son finitas. En este caso, la integral viene dada por

$$\int_E f d\mu = \int_E f_+ d\mu - \int_E f_- d\mu.$$

Si $f = g + ih$, su integral se define por

$$\int_E f d\mu = \int_E g d\mu + i \int_E h d\mu.$$

No entraremos aquí a la demostración, pero un primer resultado general para la integral de Lebesgue es que es lineal respecto a la función. Otras propiedades que se mantienen son:

- a) f, g integrables; $f \leq g \Rightarrow \int_E f d\mu \leq \int_E g d\mu$.
- b) f integrable; $|\int_E f d\mu| \leq \int_E |f| d\mu$.
- c) si $\exists g$ integrable tq. $|f| \leq g$, entonces f es integrable y $\int_E f d\mu \leq \int_E g d\mu$.

Y una cualidad importante de la integral de Lebesgue:

d) f integrable sii $|f|$ es integrable.

Este resultado es falso en general para la integral de Riemann.

Def. A.3- Dado un espacio de medida (X, Σ, μ) , definimos $L^1(X, \Sigma, \mu)$ como el espacio vectorial de funciones medibles definidas en X integrables con respecto a μ . Se escribe en ocasiones como $L^1(X)$ o L^1 si se sobreentiende por el contexto.

Teo. A.4- Teorema de Lebesgue: Sea (X, Σ, μ) espacio de medida. Consideremos una secuencia $(f_n), n \in \mathbb{N}$ de funciones de X que converge casi por todo a una función f . Si $\exists g$ integrable tq. $\forall n |f_n(x)| \leq g(x)$ casi por todo en X , entonces

i) f es integrable.

ii) $\lim_{n \rightarrow \infty} \int_E f_n d\mu = \int_E f d\mu \quad \forall E \in \Sigma$.

Pasemos a considerar ahora integrales que dependan de un parámetro. Sea un espacio de medida (X, Σ, μ) y un intervalo $(a, b) \subset \mathbb{R}$ acotado o no. Consideramos una f definida en $(a, b) \times X$ con valores en \mathbb{R} o \mathbb{C} . Asumimos que la función $x \rightarrow f(t, x)$ es integrable $\forall t \in (a, b)$. Definimos

$$I(t) = \int_X f(t, x) d\mu, \quad t \in (a, b).$$

Prop. A.5- (Continuidad) Si para casi por todo $x \in X$ la función $t \rightarrow f(t, x)$ es continua en $t_0 \in (a, b)$ y $\exists g$ tq. $\forall t$ en un entorno V de t_0 se tiene casi por todo

$$|f(t, x)| \leq g(x),$$

entonces $I(t)$ es continua en t_0 .

Prop. A.6- (Derivación) Sea V entorno de $t_0, V \subset (a, b)$ tal que se cumple

i) Casi por todo $x, x \rightarrow f(t, x)$ tiene derivada continua en V .

ii) Existe una función integrable g tq.

$$\forall t \in V, \left| \frac{\partial f}{\partial t}(t, x) \right| \leq g(x) \quad \text{c.p.t.}$$

Entonces $I(t)$ es diferenciable en t_0 , y $I'(t_0) = \int_X \frac{\partial f}{\partial t}(t_0, x) d\mu$.

Un último resultado que utilizaremos en numerosas ocasiones es el siguiente:

Teo. A.7- Teorema de Fubini. Sea $f: \mathbb{R} \times \mathbb{R} \rightarrow \overline{\mathbb{R}}$ función medible y $E \times F$ conjunto medible en $\mathbb{R} \times \mathbb{R}$.

i) f no negativa en $E \times F$ implica

$$\int_{E \times F} f(x, y) dx dy = \int_E dx \int_F f(x, y) dy = \int_F dy \int_E f(x, y) dx;$$

pudiendo ser $+\infty$.

ii) Si f es integrable en $E \times F$; entonces $x \rightarrow f(x, y)$ es integrable casi por todo y , $y \rightarrow f(x, y)$ es integrable casi para todo x , y las tres integrales anteriores son finitas e iguales.

iii) f es integrable sii

$$\int_E dx \int_F |f(x, y)| dy \quad \text{o} \quad \int_F dy \int_E |f(x, y)| dx$$

es finita.

En el sentido práctico, esto nos permite integrar escogiendo el orden, siempre que una de las dos integrales de $|f(x, y)|$ existe.

Para cerrar la sección, veamos una familia de conjuntos de funciones con la que trabajaremos.

Def. A.8- Sea $p > 0$ arbitrario, y sea I intervalo en \mathbb{R} . Entonces $L^p(I)$ denota el espacio de funciones medibles $f: I \rightarrow \mathbb{R}$ (o \mathbb{C}) para las que $|f(x)|^p$ es integrable en I .

Prop. A.9- Los espacios $L^p(I)$, $1 \leq p \leq +\infty$, son espacios normados con las normas

$$\|f\|_p = \left(\int_I |f(t)|^p dt \right)^{1/p}, \text{ si } 1 \leq p < +\infty;$$

$$\|f\|_\infty = \inf\{c \mid \text{medida}\{f(x) \geq c\} = 0\}.$$

B. Código

En este apéndice encontramos el código referenciado en la sección 10.

B.1- *Lowpass.m*

```
fileWav = input('Introducir nombre del archivo:', 's');
cutFrec = input('Introducir frecuencia máxima: ');
[x, Fs] = audioread(fileWav);
[N, K] = size(x); %N es el número de muestras
y = fft(x); %Vector de la FFT de las muestras

L = floor(N*cutFrec/Fs); %Índice del coeficiente desde el que eliminar
y(L+1:N)=zeros(1, N-L);

audiowrite('result_Lowpass.wav', ifft(y), Fs); %Resultado
```

B.2- *Compress.m*

```
window = 2048; num_components = 500; coeff_bits = 14;
fileWav = input('Introducir nombre del archivo: ', 's');
[X, Fs] = audioread(fileWav); %Leer el audio
num_win = floor(length(X)/window);
X = X(1:num_win*window); %X vector de muestras

X = reshape(X, window, num_win); %Transformado en matriz, cada ventana
es una fila
Y = dct(X); %Aplicar la DCT a cada fila

%Mantiene tan solo los coeficientes mayores
[a, I] = sort(abs(Y), 'descend');
I = I(1:num_components,:);

%res contendrá la información a guardar
res.ind = int16(I); %Índices y coeficientes
res.coeffs = int16(zeros(num_components, num_win));
for i = 1:num_win
    % store each coefficient (in [-1,1]) as an integer mapped to range
    % (-2^(coeff_bits-1), 2^(coeff_bits-1))
    res.coeffs(:, i) = int16(Y(I(:, i), i)*2^(coeff_bits-1));
end
res.window = window; %Tamaño ventana
res.coeff_bits = coeff_bits; %Numero de bits por coeficiente
res.fs = Fs; %Frecuencia de muestreo

save('compr.mat', 'res');
```

B.3- *Decompress.m*

```
load('compr.mat');
%Construimos la matriz Y con los coeficientes
num_win = size(res.coeffs, 2);
coeffs = double(res.coeffs)/(2^(res.coeff_bits-1));
Y = zeros(res.window, num_win);
for i = 1:num_win
    Y(res.ind(:,i),i) = coeffs(:,i);
end

%IDCT de cada ventana, y se concatenan
X = idct(Y);
X = reshape(X, num_win*res.window, 1);
audiowrite('decomp.wav',X, res.fs);
```

B.4- *higher.m*

```
[x,Fs]=audioread('voice.wav');
r = 0.75;

%X es la matriz de la STFT del audio
X = stft(x');

%t contiene los valores con los que interpolar
%la nueva transformada
[rows, cols] = size(X);
t = 0:r:(cols-2);
%Formamos el nuevo espectrograma e invertimos
X2 = interp(X, t);
y = istft(X2)';

audiowrite('result_Modify.wav',y,Fs); %Se escribe el resultado
```

B.5- *stft.m*

```
function STFT = stft(x)
nSamples = length(x);

%256 muestras entre cada bloque
wSize = 1024; h = wSize/4;
%Ventana a utilizar:
halfwin = (1 + cos( pi * (0:512)/512))/2;
win = zeros(1, 1024);
win(wSize/2+1:wSize) = halfwin(1:wSize/2);
win(wSize/2:-1:1) = halfwin(1:wSize/2);

%Preparamos un vector para el resultado
STFT = zeros(wSize,1+floor((nSamples-wSize)/h));
%Tomamos fragmentos de 1024 elementos, multiplicamos
%por la ventana y calculamos su fft; cogemos la primera
%mitad de los coeficientes
for b = 0:h:(nSamples-wSize)
    u = win.*x((b+1):(b+wSize));
    fftFrag = fft(u);
    STFT(:,1+b/h) = fftFrag;
end;
```

B.6- *interp.m*

```
function c = interp(b, t)
c = zeros(size(b,1), length(t));

%ph contiene la fase tras cada paso
ph = angle(b(:,1));
ocol = 1;
for tt = t
    %Las siguientes dos columnas de b
    bcols = b(:,floor(tt)+[1 2]);
    tf = tt - floor(tt);
    %Media ponderada por tf para los modulos
    bmag = (1-tf)*abs(bcols(:,1)) + tf*(abs(bcols(:,2)));
    %Calcula el cambio de fase y traslada a -pi:pi
    dp = angle(bcols(:,2)) - angle(bcols(:,1));
    dp = dp - 2 * pi * round(dp/(2*pi));
    %Columna, trasladada por la fase
    c(:,ocol) = bmag .* exp(1i*(ph));
    %Acumular fase y continuar
    ph = ph + dp;
    ocol = ocol+1;
end
```

B.7- *istft.m*

```
function x = istft(d)
wSize = 1024; h = 256;
cols = size(d,2);

%Análogo a la stft
halfwin = (1 + cos( pi * (0:512)/512))/2;
win = zeros(1, 1024);
win(wSize/2+1:wSize) = halfwin(1:wSize/2);
win(wSize/2:-1:1) = halfwin(1:wSize/2);

%x es el vector resultante
xlen = wSize + (cols-1)*h;
x = zeros(1,xlen);
for b = 0:h:(h*(cols-1))
    %Vector de la transformada de 1024 elementos
    ftFrag = d(:,1+b/h)';
    px = real(ifft(ftFrag));
    %Sumamos al valor previo, multiplicando por la ventana
    x((b+1):(b+wSize)) = x((b+1):(b+wSize))+px.*win;
end;
```

B.8- *lowpassIM.m*

```
lena512 = double(imread('lena512.bmp'));
[M,N]=size(lena512);      %Leer imagen original

F=fft2(lena512); %FFT de la imagen
S=fftshift(F); %Traslación para centrar el (0,0)

[rr cc] = meshgrid(1:M);
C = sqrt((rr-M/2).^2+(cc-M/2).^2)>M/8;
S(C) = 0; %Eliminar coeficientes altos
g=real(ifft2(ifftshift((S)))); %Reconstruir imagen resultante
imagesc(g)
axis off
imwrite(g./255,'test.bmp');
```

B.9- *recog.m*

```
%Leemos la imagen con el texto
bw = imread('text3.png');
figure; imshow(double(bw))
[M,N] = size(bw);
%Imagen de la letra 'a'
a = imread('a.png');

%C contendrá la correlación de bw y el fragmento
C = real(ifft2(fft2(bw) .* fft2(rot90(a,2),M,N)));
maxCorr = max(C(:)); %Seleccionamos el máximo
thresh = maxCorr*0.9; %Cota para determinar positivos

%Marcamos donde hemos encontrado el patrón
D = C > thresh;
se = strel('disk',3);
E = imdilate(D,se);
figure; imshow(E)
```